PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2005-094146

(43)Date of publication of application : 07.04.2005

(51)Int.Cl. H04L 9/32
G06F 17/60

(21)Application number : 2003-321966

(22)Date of filing : 12.09.2003

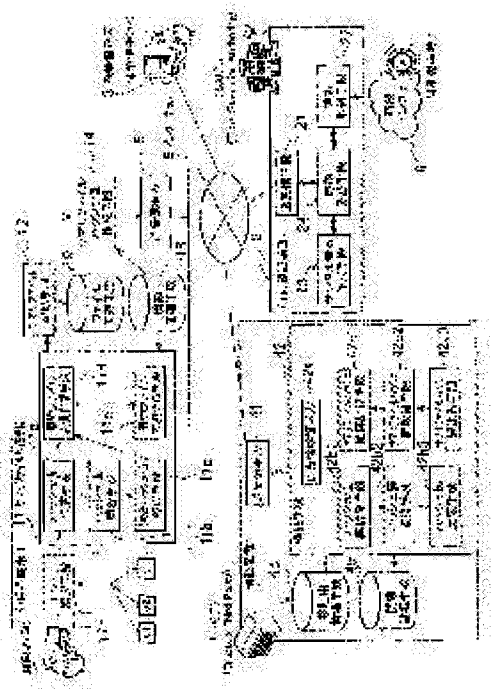
(71)Applicant : NIPPON TELEGR & TELEPH CORP <NTT>

(72)Inventor : HOTTA HIDEKAZU
ONO SATOSHI
TAKURA AKIRA

(54) PROGRAM FOR REQUESTING BULK TYPE TIME AUTHENTICATION, RECORDING MEDIUM FOR REQUESTING BULK TYPE TIME AUTHENTICATION, VERIFICATION APPARATUS, VERIFICATION METHOD, VERIFICATION PROGRAM, AND VERIFICATION RECORDING MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a method wherein hash values of a plurality of files being proof objects of time authentication are collected into one to create one file and time confirmation is requested totally by the whole file, that can easily prove an unrevised part to be valid even when part of entry (digital data) related to the proof object is revised when a time authentication certificate is issued and reduce an amount of data required to be transmitted to a third party for which the propriety of the time authentication certificate of part of digital data is to be shown.



SOLUTION: A file (summary file) having characteristics as shown below is used for a file being a proof object of time authentication; (1) the file is described in a particular description language such as the XML understandable by people, (2) the name of entry e is displayed, (3) hash data of the entry e are displayed, wherein the hash data indicate the hash algorithm in use, calculated hash values, and a hash key depending on case, (4)

(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2005-94146
(P2005-94146A)
(43)公開日　平成17年4月7日(2005.4.7)

(51)Int.Cl.[7]	F I		テーマコード（参考）
H04L 9/32	H04L 9/00	675Z	5J104
G06F 17/60	G06F 17/60	140	
	H04L 9/00	675D	

審査請求 未請求　請求項の数 13　OL　（全 32 頁）

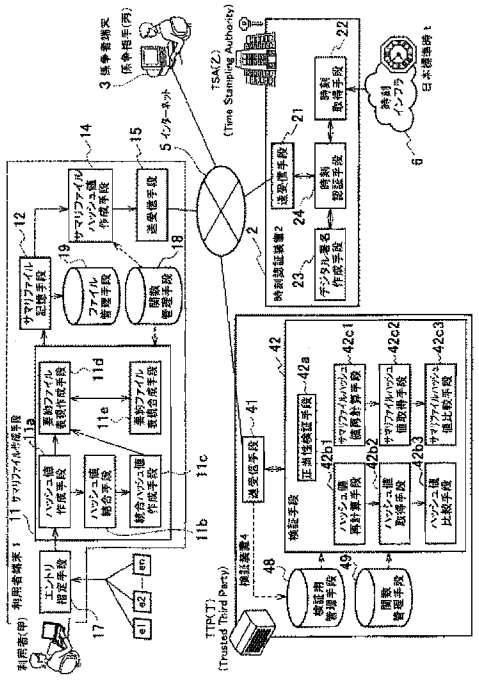
(21)出願番号　特願2003-321966 (P2003-321966)
(22)出願日　平成15年9月12日 (2003.9.12)

(71)出願人　000004226
日本電信電話株式会社
東京都千代田区大手町二丁目３番１号
(74)代理人　100083806
弁理士　三好　秀和
(72)発明者　堀田　英一
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内
(72)発明者　小野　諭
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内
(72)発明者　田倉　昭
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内

Fターム(参考)　5J104 AA11　LA05

(54)【発明の名称】バルク型時刻認証要求用プログラム、及びバルク型時刻認証要求用記録媒体、並びに、検証装置、検証方法、検証用プログラム、及び検証用記録媒体

(57)【要約】
【課題】時刻認証の証明対象となる複数ファイルのハッシュ値を１つにまとめて１つのファイルを作成し、このファイル全体で一括して時刻認証要求を行う方法において、時刻認証証明書が発行された後に、証明対象に係るエントリ（デジタルデータ）の一部に変更が生じても、変更が生じていない部分は有効であると容易に証明することができることと一部のデジタルデータに対して時刻認証証明書の正当性を第三者に示すために送付する必要のあるデータの量を著しく削減するを目的とする。
【解決手段】上記１つのファイルを以下に示す特徴のファイル（サマリファイル）とする。(1)人間が理解できるＸＭＬ等の特定の記述言語により記述。(2)エントリｅの名称を表示。(3)エントリｅのハッシュデータを表示。ハッシュデータは、使用したハッシュアルゴリズム、計算したハッシュ値、場合によってはハッシュキーが示されている。(4)木構造をテキストファイルとして表現。
【選択図】　図１



【特許請求の範囲】
【請求項1】
電子フォルダ又は電子ファイルとしての複数のエントリのハッシュ値から新たな１つのハッシュ値を作成し、当該１つのハッシュ値を時刻認証の証明対象として時刻証明機関へ時刻認証要求を行うためのバルク型時刻認証要求用のプログラムであって、
コンピュータが、前記エントリ毎に定義されかつ当該エントリに適用するエントリ用ハッシュ関数及びサマリファイルに適用するサマリファイル用ハッシュ関数を管理する関数管理手段と、前記サマリファイルを管理するファイル管理手段とを有した状態において、
前記複数のエントリの各々に対して前記関数管理手段で管理しているエントリ用ハッシュ関数を適用して、各エントリのハッシュ値を作成するハッシュ値作成手段と、
前記ハッシュ値作成手段によって作成した各エントリのハッシュ値と当該各エントリを特定する識別子をそれぞれ関連づけて、前記エントリ毎の要約ファイル表現を作成する要約ファイル表現作成手段と、
前記要約ファイル表現作成手段によって作成したエントリ毎の要約ファイル表現を結合する要約ファイル表現合成手段と、
前記要約ファイル表現合成手段によって要約ファイル表現を結合した結果として作成したサマリファイルを、前記ファイル管理手段に記憶するサマリファイル記憶手段と、
前記サマリファイル管理手段に記憶すべきサマリファイルに対して前記関数管理手段で管理しているサマリファイル用ハッシュ関数を適用して、前記サマリファイルのハッシュ値を作成するサマリファイルハッシュ値作成手段と、
を前記コンピュータに機能させることにより、前記作成したサマリファイルのハッシュ値を時刻認証の証明対象として時刻証明機関への時刻認証要求を可能とするためのバルク型時刻認証要求用プログラム。
【請求項2】
前記エントリが電子フォルダの場合において、
前記ハッシュ値作成手段は、
前記電子フォルダに階層的に格納されている電子ファイル又は電子サブフォルダに係るエントリに基づいた要約ファイル表現から導出され、少なくとも前記電子フォルダに格納されているファイルに係る※各エントリのハッシュ値を含むデジタルデータに対して前記エントリ用ハッシュ関数を適用することによって、前記電子フォルダに係るエントリのハッシュ値を作成することが可能であり、
更に、前記電子フォルダに係るエントリのハッシュ値は、少なくとも前記電子フォルダに含まれる前記各エントリの識別子、当該エントリが電子ファイルである場合には当該電子ファイルの内容としてのデジタルデータに依存して決定されるものであることを特徴とする請求項１に記載のバルク型時刻認証要求用プログラム。
【請求項3】
前記サマリファイルハッシュ値作成手段は、前記サマリファイルに含まれる１個以上の要約ファイル表現の構成要素のうちで少なくともハッシュ値の部分を含むようなデジタルデータに対して前記サマリファイル用ハッシュ関数を適用することにより、サマリファイルのハッシュ値を作成することを特徴とする請求項１に記載のバルク型時刻認証要求用プログラム。
【請求項4】
請求項１乃至３に記載のバルク型時刻認証要求用プログラムであって、更に、
サマリファイルに該エントリの要約ファイル表現を含めることにより前記時刻認証の証明対象となるエントリを指定するエントリ指定手段を有することを特徴とするバルク型時刻認証要求用プログラム。
【請求項5】
請求項４に記載のバルク型時刻認証要求用プログラムであって、更に、
指定されたエントリに対してグループ化し，各グループに識別子を定義し，それらのグ

ループをエントリと見なすことによりグループ間の入れ子構造を定義するグループ定義手段を有することを特徴とするバルク型時刻認証要求用プログラム。

【請求項6】

請求項１乃至５に記載のバルク型時刻認証要求用プログラムを記録した、コンピュータ読み取り可能なバルク型時刻認証要求用記録媒体。

【請求項7】

電子フォルダ又は電子ファイルとしての複数のエントリのハッシュ値から新たな１つのハッシュ値を作成し、利用者が前記１つのハッシュ値を時刻認証の証明対象として時刻証明機関へ時刻認証要求を行うことで時刻認証証明書を取得した後に、前記利用者が取得した時刻認証証明書の正当性についての検証を行う検証装置であって、

前記複数のエントリ、前記利用者端末で作成されたサマリファイル、及び時刻認証証明書のデータを取得して管理しておく検証用管理手段と、

前記エントリに適用するエントリ用ハッシュ関数及びサマリファイルに適用するサマリファイル用ハッシュ関数を管理する関数管理手段と、

前記検証用管理手段に管理しておいた時刻認証証明書が改ざんされてなく正しいことを検証する正当性検証手段と、

前記検証用管理手段に管理しておいた複数のエントリのうち検証の対象となるものに対して前記関数管理手段で管理しているエントリ用ハッシュ関数を適用して、各エントリのハッシュ値を再計算するハッシュ値再計算手段と、

前記検証用管理手段に管理しておいたサマリファイルを構成する要約ファイル表現中の対応する元のエントリのハッシュ値を取得するハッシュ値取得手段と、

前記ハッシュ値再計算手段によって再計算した前記エントリのハッシュ値と、前記ハッシュ値取得手段によって取得したハッシュ値とが同じであるか否かを比較するハッシュ値比較手段と、

前記検証用管理手段に管理しておいたサマリファイルに対して前記関数管理手段で管理しているサマリファイル用ハッシュ関数を適用して、前記サマリファイルのハッシュ値を再計算するサマリファイルハッシュ値再計算手段と、

前記検証用管理手段に管理しておいた時刻認証証明書に含まれる元のサマリファイルのハッシュ値を取得するサマリファイルハッシュ値取得手段と、

前記サマリファイルハッシュ値再計算手段で再計算したサマリファイルのハッシュ値と前記サマリファイルハッシュ値取得手段で取得した元のサマリファイルのハッシュ値とが同じであるか否かを比較するサマリファイルハッシュ値比較手段と、

を有することを特徴とする検証装置。

【請求項8】

電子フォルダ又は電子ファイルとしての複数のエントリのハッシュ値から新たな１つのハッシュ値を作成し、利用者が前記１つのハッシュ値を時刻認証の証明対象として時刻証明機関へ時刻認証要求を行うことで時刻認証証明書を取得した後に、前記利用者が取得した時刻認証証明書の正当性についての検証を行う検証装置であって、

前記複数のエントリ、前記利用者端末で作成されたサマリファイル、及び時刻認証証明書のデータを取得して管理しておく検証用管理手段と、

前記エントリに適用するエントリ用ハッシュ関数及びサマリファイルに適用するサマリファイル用ハッシュ関数を管理する関数管理手段と、

前記検証用管理手段に管理しておいた時刻認証証明書が改ざんされてなく正しいことを検証する正当性検証手段と、

前記検証用管理手段に管理しておいた複数のエントリのうち検証の対象となるものに対して前記関数管理手段で管理しているエントリ用ハッシュ関数を適用して、各エントリのハッシュ値を再計算するハッシュ値再計算手段と、

前記検証の対象となる各エントリについて、その識別子とハッシュ値に加えて、上記サマリファイルのハッシュ値を計算するのに必要なデータを当該のエントリに対する検証用補完データと定義し、さらに、当該のエントリの識別子、そのハッシュ値、そのハッシュ

値を計算するためのハッシュ関数の識別子、及び当該のエントリに対する検証用補完データからなるデータを当該エントリの個別化サマリファイルと定義すると、検証の対象となる各エントリに対する個別化サマリファイルから上記サマリファイルのハッシュ値を計算する個別化サマリファイル利用型サマリファイル・ハッシュ値再計算手段と、

前記検証用管理手段に管理しておいた時刻認証証明書に含まれる元のサマリファイルのハッシュ値を取得するサマリファイルハッシュ値取得手段と、

前記個別化サマリファイル利用型サマリファイル・ハッシュ値再計算手段で再計算したサマリファイルのハッシュ値と前記サマリファイルハッシュ値取得手段で取得した元のサマリファイルのハッシュ値とが同じであるか否かを比較するサマリファイルハッシュ値比較手段と、

を有することを特徴とする検証装置。

【請求項9】

電子フォルダ又は電子ファイルとしての複数のエントリのハッシュ値から新たな１つのハッシュ値を作成し、利用者が前記１つのハッシュ値を時刻認証の証明対象として時刻証明機関へ時刻認証要求を行うことで時刻認証証明書を取得した後に、前記利用者が取得した時刻認証証明書の正当性についての検証を行う検証装置を用いた検証方法であって、

前記検証装置が、前記複数のエントリ、前記利用者端末で作成されたサマリファイル、及び時刻認証証明書のデータを取得して管理しておく検証用管理手段と、前記エントリに適用するエントリ用ハッシュ関数及びサマリファイルに適用するサマリファイル用ハッシュ関数を管理する関数管理手段とを有した状態において、

前記検証装置は、

前記検証用管理手段に管理しておいた時刻認証証明書が改ざんされてなく正しいことを検証する正当性検証ステップと、

前記検証用管理手段に管理しておいた複数のエントリのうち検証の対象となるものに対して前記関数管理手段で管理しているエントリ用ハッシュ関数を適用して、各エントリのハッシュ値を再計算するハッシュ値再計算ステップと、

前記検証用管理手段に管理しておいたサマリファイルを構成する要約ファイル表現中の対応する元のエントリのハッシュ値を取得するハッシュ値取得ステップと、

前記ハッシュ値再計算ステップによって再計算した前記エントリのハッシュ値と、前記ハッシュ値取得ステップによって取得したハッシュ値とが同じであるか否かを比較するハッシュ値比較ステップと、

前記検証用管理手段に管理しておいたサマリファイルに対して前記関数管理手段で管理しているサマリファイル用ハッシュ関数を適用して、前記サマリファイルのハッシュ値を再計算するサマリファイルハッシュ値再計算ステップと、

前記検証用管理手段に管理しておいた時刻認証証明書に含まれる元のサマリファイルのハッシュ値を取得するサマリファイルハッシュ値取得ステップと、

前記サマリファイルハッシュ値再計算手段で再計算したサマリファイルのハッシュ値と前記サマリファイルハッシュ値取得ステップで取得した元のサマリファイルのハッシュ値とが同じであるか否かを比較するサマリファイルハッシュ値比較ステップと、

を実行することを特徴とする検証方法。

【請求項10】

電子フォルダ又は電子ファイルとしての複数のエントリのハッシュ値から新たな１つのハッシュ値を作成し、利用者が前記１つのハッシュ値を時刻認証の証明対象として時刻証明機関へ時刻認証要求を行うことで時刻認証証明書を取得した後に、前記利用者が取得した時刻認証証明書の正当性についての検証を行う検証装置を用いた検証方法であって、

前記検証装置が、前記複数のエントリ、前記利用者端末で作成されたサマリファイル、及び時刻認証証明書のデータを取得して管理しておく検証用管理手段と、前記エントリに適用するエントリ用ハッシュ関数及びサマリファイルに適用するサマリファイル用ハッシュ関数を管理する関数管理手段とを有した状態において、

前記検証装置は、

前記検証用管理手段に管理しておいた複数のエントリのうち検証の対象となるものに対して前記関数管理手段で管理しているエントリ用ハッシュ関数を適用して、各エントリのハッシュ値を再計算するハッシュ値再計算ステップと、

前記検証の対象となる各エントリについて、その識別子とハッシュ値に加えて、上記サマリファイルのハッシュ値を計算するのに必要なデータを、当該のエントリに対する検証用補完データと定義し、さらに、当該のエントリの識別子、そのハッシュ値、そのハッシュ値を計算するためのハッシュ関数の識別子、及び当該のエントリに対する検証用補完データからなるデータを当該エントリの個別化サマリファイルと定義すると、検証の対象となる各エントリに対する個別化サマリファイルから上記サマリファイルのハッシュ値を計算する個別化サマリファイル利用型サマリファイル・ハッシュ値再計算ステップと、

前記検証用管理手段に管理しておいた時刻認証証明書が改ざんされてなく正しいことを検証する正当性検証ステップと、

前記検証用管理手段に管理しておいた時刻認証証明書に含まれる元のサマリファイルのハッシュ値を取得するサマリファイルハッシュ値取得ステップと、

前記個別化サマリファイル利用型サマリファイル・ハッシュ値再計算ステップで再計算したサマリファイルのハッシュ値と前記サマリファイルハッシュ値取得ステップで取得した元のサマリファイルのハッシュ値とが同じであるか否かを比較するサマリファイルハッシュ値比較ステップと、

を実行することを特徴とする検証方法。

【請求項11】

請求項９に記載の検証装置に、上記各ステップを実行させることを特徴とした検証用プログラム。

【請求項12】

請求項１０に記載の検証装置に、上記各ステップを実行させることを特徴とした検証用プログラム。

【請求項13】

請求項１１及び請求項１２のいずれか1項に記載のプログラムを記録したことを特徴とした、コンピュータ読み取り可能な検証用記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明はデジタル文書等のデジタルデータに時刻印を押すサービスにおいて、デジタルデータが時刻印を押された時点以降において変更されてなく、かつ確かに時刻印が押された時点で対象とするデジタルデータが存在していたことを証明することを可能とする時刻認証方法に関し、特に、電子フォルダ又は電子ファイルとしての複数のエントリのハッシュ値から新たな１つのハッシュ値を作成し、当該１つのハッシュ値を時刻認証の証明対象として時刻証明機関へ時刻認証要求を行うためのバルク(BULK)型時刻認証に関する。

【背景技術】

【0002】

例えば、米国における先発明主義に基づく特許制度の下では日付の入った研究ノートが優先権を証明する証拠として用いることが可能であり、さらに日付が付けられた家計簿は確定申告における支出記録として使うことができることが知られている。一方、パソコンが日常的に使用されるようになるにつれ、研究ノートや家計簿などの日常記録をパソコンを用いて行うことが、ごく一般的になってきている。

【0003】

しかし、このようなパソコン上での電気的、デジタル的な記録による文書等は容易に書き換えることができることから、記録媒体としての紙を用いて書かれた記録とは異なり、記録日時を含め記録内容を第三者に証明することができないという問題を有していた。そのため、従来は、ＴＳＡ(Time-Stamp Authority：時刻証明機関)が、時刻認証証明書を発行する時刻認証サービスを行うことで、デジタルデータの時刻証明の問題を解決していた。

【0004】
更に、近年は、図１５に示すように、利用者（甲）がＴＳＡ（乙）に、時刻認証の証明対象となるデジタルデータに対して時刻認証証明書を作成してもらい、その後、係争相手（丙）に対して時刻認証証明書の正当性を証明する必要が出てきた場合には、ＴＴＰ（Trusted Third Party：信頼される第三者機関）（丁）に、その正当性の検証を依頼することで、紛争のより迅速な解決を図ることが可能となった。この場合、利用者（甲）は、利用者端末２１０からインターネット５を介してＴＳＡ（乙）の時刻認証装置２に時刻認証要求を行い、時刻認証装置２で時刻認証証明書を作成して利用者端末２１０に送信する。そして、利用者（甲）が係争相手（丙）との間で、時刻認証証明書の正当性を証明する必要が出てきた場合には、利用者端末２１０からインターネット５を介してＴＴＰ（丁）の検証装置２４０に時刻証明検証要求を行い、検証装置２４０で正当性を検証して、検証結果を利用者端末１１０や係争者端末３に送信する。
【0005】
更に、最近、利用者（甲）がコンピュータ（計算機）のファイルシステムの特定の電子フォルダｆに含まれる複数のファイルに係るデジタルデータｇ１、ｇ２、・・・ｇｎ（ｎは正数、以下、総称を「ｇ」とする）をまとめ、一括して時刻認証証明書を取得するような時刻認証方法が提案されている（特許文献１参照）。この方法は、１つにまとめた「固まり」という意味のバルク(bulk)から、バルク型時刻認証方法と呼ばれている。
【0006】
ここで、図１６を用い、上記従来のバルク型時刻認証方法について説明する。尚、図１６は、従来のバルク型時刻認証方法における登場主体の関係、通信インフラストラクチャ(Infrastructure)、及び各装置の構成を示した図である。
【0007】
図１６に示すように、利用者（甲）は、デジタルデータＧを作成し、ＴＳＡ（乙）に時刻認証証明書の作成を依頼する者であり、パソコン等の利用者端末１１０からインターネット５を介してＴＳＡ（乙）の時刻認証装置２に時刻認証要求を行うことができる。ＴＳＡ（乙）は、時刻インフラストラクチャ６を介して取得した日本標準時ｔに基づいて、時刻認証証明書を作成する者である。係争相手（丙）は、利用者（甲）が自己のデジタルデータの作成日時等に関して争う当事者の一方であり、パソコン等の係争者端末３からインターネット５を介して利用者端末１１０又は時刻認証装置２に対し、デジタルデータの送信要求を行うことができる。ＴＴＰ（丁）は、信頼される第三者機関として、中立性と客観性をもち、利用者（甲）と係争相手（丙）の紛争解決に役立てるために時刻認証証明書の正当性を検証する機関であり、検証装置１４０によりインターネット５を介して利用者端末１１０からの時刻証明検証要求を受け付けて時刻認証証明書の正当性の検証を行うと共に、インターネット５を介して検証結果を利用者端末１１０又は係争者端末３に送信することができる。
【0008】
次に、利用者端末１１０、時刻認証装置２、及び検証装置１４０について説明する。尚、利用者端末１１０には、この利用者端末１１０を図１６に示す各手段として機能させるための時刻認証要求用プログラムが記憶されている。また、時刻認証装置２には、この時刻認証装置２を図１６に示す各手段として機能させるための時刻認証用プログラムが記憶されている。更に、検証装置１４０には、この検証装置１４０を図１６に示す各手段として機能させるための正当性検証用プログラムが記憶されている。
【0009】
また、利用者端末１１０は、上記時刻認証要求用プログラムを実行することによって、図１６に示すように、ハッシュ値作成手段１１１、ハッシュ値結合手段１１２、統合ハッシュ値作成手段１１３、及び、送受信手段１１４を有するに至っている。
【0010】
まず、ハッシュ値作成手段１１１は、テキスト情報、画像情報、音声情報等のデジタルデータＧのうちで、時刻認証の証明対象となるフォルダｆ内の複数のデジタルデータｇ毎

にハッシュ関数h1を適用して、各ハッシュ値h1(ｇ１)，h1(ｇ２)，、、、、，h1(ｇｎ)（以下、これらをまとめて表す場合は「h1(ｇ)」とする）を作成することができる。このハッシュ値は、別名「ダイジェスト」ともいう。このようなハッシュ値h1(ｇ)の作成目的は、デジタルデータｇのような不定長のデータを所定の手順に従って変換して、一定の長さ（例えば、２０byte）のハッシュ値h1(ｇ)に圧縮することで、デジタルデータｇに署名を行う代わりに、そのデジタルデータｇのハッシュ値に署名することで、計算や送信の効率を向上させることである。また、安全なハッシュ関数の条件は、ハッシュ値から入力を推定することが困難であること、及び２つの異なるデジタルデータのハッシュ値が一致する確率が極めて小さいことである。例えば、ハッシュ関数として、MD5、SHA-1等がある。ハッシュ関数は、Ｃ言語のライブラリ等として広く流通しており、仕様が公開されているため、各人が自分でプログラム化することも比較的容易である。このハッシュ関数は、公開されている仕様通りに動作することで、ハッシュ値の送受信者間で合意されている必要がある。

【0011】

次に、ハッシュ値結合手段１１２は、ハッシュ値作成手段１１で作成された複数のハッシュ値h1（ｇ）を連接することにより結合した結果として、次式を得る。

【0012】

h1(ｇ１)‖h1(ｇ２)‖・・・‖h1(ｇｎ)

ここで、ハッシュ値の連接とは、ハッシュ値をバイト（byte）列とみて、単純につないで行くことを示す。例えば、それぞれ２０byteのハッシュ値h1(ｇ１)とh1(ｇ２)を連接すると、合計で４０byteのバイト列となり、前半の２０byteはh1(ｇ１)と同じで、後半の２０byteはh1(ｇ２)と同じとなるような、単純なバイト列のつながりのことである。

【0013】

次に、統合ハッシュ値作成手段１１３は、ハッシュ値結合手段１１２によって得られた結合結果の全体に、統合用ハッシュ関数h2を適用して、次式の統合ハッシュ値を得る。

【0014】

h2(h1(ｇ１)‖h1(ｇ２)‖・・・‖h1(ｇｎ))

尚、この統合ハッシュ値を簡略化して表すために、以下「ｄ」で表す。

【0015】

次に、利用者（甲）は、図１７に示すように、利用者端末１１０からインターネット５を介して統合ハッシュ値ｄをＴＳＡ（乙）の時刻認証装置２へ送信することで、時刻認証要求を行う（ステップＳ１０１）。

【0016】

一方、時刻認証装置２は、時刻認証用プログラムを実行することによって、図１６に示すように、送受信手段２１、時刻取得手段２２、時刻認証手段２３、及び、デジタル署名作成手段２４を有するに至っている。

【0017】

まず、送受信手段２１は、インターネット５を介して上記利用者端末１１０から送られてきた統合ハッシュ値ｄを受信する。また、時刻取得手段２２は、時刻インフラストラクチャ６から日本標準時ｔを取得している。

【0018】

次に、デジタル署名作成手段２３は、送受信手段２１で受信した統合ハッシュ値ｄのデータに、時刻取得手段２２で取得した時刻ｔ、及び所定の附加情報を結合し時刻認証証明書における署名対象デジタルデータ TSTI(d, t)とし、この結合した全体に対してデジタル署名 sig(SK, TSTI(d, t))を作成する。このデジタル署名 sig(SK, TSTI(d, t)) は、ＴＳＡ（乙）の署名鍵（秘密鍵）SKによるデジタルデータ TSTI(d, t) に対するデジタル署名を示したものである。

【0019】

次に、時刻認証手段２４は、上記統合ハッシュ値ｄと時刻ｔ を含む署名対象デジタルデータ TSTI(d, t)、及び、デジタル署名 sig(SK, TSTI(d, t))を含む時刻認証証明書 TST(

sig(SK, TSTI(d, t)), TSTI(d, t)) を作成する。この時刻認証証明書 TST(sig(SK, TSTI(d, t)), TSTI(d, t)) は、署名付きデジタルデータとしての時刻認証証明書を示したものである。これにより、ＴＳＡ（乙）は、図１７に示すように、時刻認証装置２から利用者（甲）の利用者端末１１０へ、時刻認証証明書 TST(sig(SK, TSTI(d, t)), TSTI(d, t)) を送信することで、時刻証明応答を行う（ステップＳ１０２）。

【0020】

また、検証装置１４０は、検証用プログラムを実行することによって、図１６に示すように、送受信手段１４１、及び、検証手段１４２を有するに至っている。尚、署名鍵SKと、この署名鍵SKと対になる公開鍵PKは、何らかの手段でＴＴＰ（丁）に知られている。

【0021】

そして、利用者（甲）が、デジタルデータｇの時刻証明を係争相手（丙）に証明する必要が出てきた場合には、図１７に示すように、利用者（甲）は、利用者端末１１０からインターネット５を介してＴＴＰ（丁）の検証装置１４０に、証明対象である複数のデジタルデータｇを格納したフォルダｆ〔ｇ〕、及び時刻認証証明書 TST1 を送信することで、時刻証明検証要求を行う（ステップＳ１０３）。

【0022】

次に、検証装置１４０では、送受信手段１４１によって、フォルダｆ〔ｇ〕及び時刻認証証明書 TST1を受信し、検証用記憶手段１４９に記憶しておく。そして、検証手段１４２は、更に、統合ハッシュ値再計算手段１４３、デジタル署名検証手段１４４、及び、統合ハッシュ値比較手段１４５を有しており、各手段によって以下に示す手順で検証を行う。

【0023】

まず、検証時の入力対象として、上記フォルダｆ〔ｇ〕の名称ｆｎ、フォルダｆ〔ｇ〕に含まれるデジタルデータｇ、及び、フォルダｆ〔ｇ〕に対して既に取得している時刻認証証明書 TST1 が入力されると、統合ハッシュ値再計算手段１４３によって、検証用記憶手段１４９からフォルダｆ〔ｇ〕の名称ｆｎに対応したデジタルデータｇを読み出し、上記利用者端末１１０で統合ハッシュ値ｄを作成した手順と同じ手順で、統合ハッシュ値ｄ’を再度計算する。

【0024】

次に、デジタル署名検証手段１４４により、時刻認証証明書 TST1 から署名対象デジタルデータ TSTI(d, t) と、デジタル署名 sig(SK, TSTI(d, t)) を抽出し、公開の検証鍵 PK を用いてデジタル署名 sig(SK, TSTI(d, t)) が、署名対象デジタルデータ TSTI(d, t) に対してなされたものであることを検証する。

【0025】

次に、統合ハッシュ値比較手段１４５により、上記再計算した統合ハッシュ値ｄ’と、上記時刻認証の署名対象デジタルデータTSTI(d, t) から抽出したハッシュ値 ｄ を比較し、ｄ’＝ｄか否かを検証する。この場合、署名対象デジタルデータTSTI(d, t)の書式は定まっているため TSTI(d, t) からｄを取り出すことかできる。これにより、ｄ’＝ｄが成り立てば、検出結果として「正当（ＯＫ）」を出力し、ｄ’＝ｄが成り立たなければ、検出結果として「不当（ＮＧ）」を出力する。

【0026】

このように、従来のバルク型時刻認証方法によれば、フォルダｆ〔ｇ〕を一括してまとめた状態で、単一の時刻認証証明書TST(sig(SK, TSTI(d, t)), TSTI(d, t))の要求を行うことで、ＴＳＡ（乙）が時刻認証証明書の発行を単一で済むと共に、利用者（甲）も単一の時刻認証証明書 TST(sig(SK, TSTI(d, t)), TSTI(d, t))を管理するだけで済むという効果を奏する。

【0027】

関連する他の従来技術として、構造を持ったデジタル文書とそのデジタル署名を一体化して記述するためのマークアップ言語であるSDML (Signed Document Markup Langauge) がある（非特許文献１参照）。この言語の本来の使用目的は、デジタル署名の対象となる

メッセージの内容を該言語で構造化して文字列として表現し、該文字列とそれに対するデジタル署名を含むような構造化されたデジタル文書を作成することにある。SDMLは、メッセージの内容の代わりに時刻証明の対象となるデジタル・データ（電子ファイル等）の識別子（パス名）とそのハッシュ値の組を用い、かつSDMLで使われるデジタル署名として信頼される第三者機関によって発行される時刻印つきのデジタル署名を用いることにより、本発明と同じ時刻証明の目的のために用いることができると考えられる。この際、複数のデジタル・データをグループ化し時刻証明書を取得するには、当該の各デジタル・データの識別子とハッシュ値の組を SDML におけるデータ・ブロックとして扱い、これら複数のデータ・ブロックに対して、信頼される第三者機関によって発行される時刻印つきのデジタル署名を取得してシグニチャ・ブロックを生成し、これを時刻証明書とする。また１つのデジタル・データに対して時刻証明書の検証を行うには、上記のシグニチャ・ブロックの検証をSDML で定められた手順に従って行うと同時に、当該のデジタル・データの識別子とそのハッシュ値の組が対応するSDMLのデータ・ブロックの内容と一致することの検証を行う。しかし、たとえこのような拡張を行ってもSDML においてはブロックはネストを許さないため、多段の階層を持ったデジタル・データの集まりに対して、時刻証明を実現する方法は文献からは明らかでない。

【0028】

SDMLをベースに、多段に階層化された複数のデジタル・データに対して時刻証明を実現する方法としては、この多段に階層化された複数のデジタル・データの階層構造を無視し、複数のデジタル・データの単純な集まりとして扱い、前記の方法を適用することが考えられる。しかし、この方法を用いた場合、第三者がある１つのデジタル・データに対して時刻証明書の検証を行うためには、当該のデジタル・データと共に時刻証明書を検証者に提供する必要があるが、時刻証明書のデータ量は時刻証明の対象であるデジタル・データの数 N に比例して大きくなるという問題がある。一方、多段に階層化されたサマリファイルを用いる本発明においては、検証者が１つのデジタル・データに対して時刻証明書の検証を行うために必要となる最小限の情報を元のサマリファイルから抽出して個別化サマリファイルを作成し、これを用いて時刻証明書の検証をすることが出来る（後述の第４の実施の形態を参照）。個別化サマリファイルのデータ量は、各階層における分岐の数をmとすると、$m \cdot \log_m N$ に比例する。従って、個別化サマリファイルを利用する本発明の方式においては、N が大きいとき、第三者が１つのデジタル・データに対して時刻証明書の検証を行う際の通信量を著しく削減すると同時に、検証を行うために必要な情報のみを開示することによりプライバシに係わる可能性のある情報の開示を最小限に抑えるという利点がある。さらに、検証時に扱うデータ量が小さくなることにより、検証のための処理時間が短くなるという利点がある。

【特許文献1】特開２００１－１４２３９８号公報

【非特許文献1】SDML-Signed Document Markup Language Specifications Version 2.0 (http://www.fstc.org/)

【発明の開示】

【発明が解決しようとする課題】

【0029】

しかしながら、従来のバルク型時刻認証方法では、一括して１つにまとめたフォルダｆ〔ｇ〕の全体の統合ハッシュ値ｄに対して時刻認証証明書 TST1を発行しているため、フォルダｆ〔ｇ〕のうちの一部のデジタルデータ（例えば、ｇ１）に変更が生じた場合には、時刻認証証明書の全体が無効になってしまうという問題が生じていた。また、一部のデジタルデータに対して時刻認証証明書の正当性を第三者に示すために、大量のデータを、第三者に送付しなければならないという問題が生じていた。

本発明は、上述した事情に鑑みてなされたもので、時刻認証証明書が発行された後に、証明対象に係るデジタルデータの一部に変更が生じても、変更が生じていない部分は有効であると証明することができ、また一部のデジタルデータに対して時刻認証証明書の正当性を第三者に示すために送付する必要のあるデータの量を著しく削減する発明を公開する

ことを目的とする。

【課題を解決するための手段】

【0030】

上記目的を達成するため、請求項１に係る発明は、電子フォルダ又は電子ファイルとしての複数のエントリのハッシュ値から新たな１つのハッシュ値を作成し、当該１つのハッシュ値を時刻認証の証明対象として時刻証明機関へ時刻認証要求を行うためのバルク型時刻認証要求用のプログラムであって、コンピュータが、前記エントリ毎に定義されかつ当該エントリに適用するエントリ用ハッシュ関数及びサマリファイルに適用するサマリファイル用ハッシュ関数を管理する関数管理手段と、前記サマリファイルを管理するファイル管理手段とを有した状態において、前記複数のエントリの各々に対して前記関数管理手段で管理しているエントリ用ハッシュ関数を適用して、各エントリのハッシュ値を作成するハッシュ値作成手段と、前記ハッシュ値作成手段によって作成した各エントリのハッシュ値と当該各エントリを特定する識別子をそれぞれ関連づけて、前記エントリ毎の要約ファイル表現を作成する要約ファイル表現作成手段と、前記要約ファイル表現作成手段によって作成したエントリ毎の要約ファイル表現を結合する要約ファイル表現合成手段と、前記要約ファイル表現合成手段によって要約ファイル表現を結合した結果として作成したサマリファイルを、前記ファイル管理手段に記憶するサマリファイル記憶手段と、前記サマリファイル管理手段に記憶すべきサマリファイルに対して前記関数管理手段で管理しているサマリファイル用ハッシュ関数を適用して、前記サマリファイルのハッシュ値を作成するサマリファイルハッシュ値作成手段と、を前記コンピュータに機能させることにより、前記作成したサマリファイルのハッシュ値を時刻認証の証明対象として時刻証明機関への時刻認証要求を可能とするためのバルク型時刻認証要求用プログラムである。

【0031】

ここで、「プログラム」とは、コンピュータによる処理に適した命令の順番付けられた列からなるものをいい、コンピュータのＨＤ(Hard Disk)、ＣＤ－ＲＷ等にインストールされているものや、ＣＤ－ＲＯＭ、ＤＶＤ、ＦＤ、半導体メモリ、コンピュータのＨＤＤ等の各種記録媒体に記録されているものや、インターネット等の外部ネットワークを介して配信されるものも含まれる（以下、同様）。

【0032】

また、「電子フォルダ」には、ある電子フォルダに含まれるもう１つの電子フォルダであるような電子サブフォルダが含まれる（以下、同様）。

【0033】

請求項２に係る発明は、前記エントリが電子フォルダの場合において、前記ハッシュ値作成手段は、前記電子フォルダに格納されている電子ファイル又は電子サブフォルダに係るエントリに基づいた要約ファイル表現から導出され、少なくとも前記電子フォルダに格納されているファイルに係る※各エントリのハッシュ値を含むデジタルデータに対して前記エントリ用ハッシュ関数を適用することによって、前記電子フォルダに係るエントリのハッシュ値を作成することが可能であり、更に、前記電子フォルダに係るエントリのハッシュ値は、少なくとも前記電子フォルダに含まれる前記各エントリの識別子、当該エントリが電子ファイルである場合には当該電子ファイルの内容としてのデジタルデータに依存して決定されるものであることを特徴とする請求項１に記載のバルク型時刻認証要求用プログラムである。

【0034】

ここで、「デジタルデータ」が、エントリ用ハッシュ関数を適用することが可能であればよく、例えば、文字列、バイト（byte）列、ビット（bit）列が挙げられる。

請求項３に係る発明は、前記サマリファイルハッシュ値作成手段は、前記サマリファイルの要約ファイル表現の構成要素のうちで少なくともハッシュ値の部分を含むようなデジタルデータに対して前記サマリファイル用ハッシュ関数を適用することにより、サマリファイルのハッシュ値を作成することを特徴とする請求項１に記載のバルク型時刻認証要求用プログラムである。

【0035】
請求項４に係る発明は、請求項１乃至３に記載のバルク型時刻認証要求用プログラムであって、更に、前記時刻認証の証明対象となるエントリ（即ち前記サマリファイルに含めるエントリ）を指定するエントリ指定手段を有することを特徴とするバルク型時刻認証要求用プログラムである。
【0036】
請求項５に係る発明は、請求項４に記載のバルク型時刻認証要求用プログラムであって、更に、指定されたエントリに対してグループ化し，各グループに識別子を定義し，それらのグループをエントリと見なすことによりグループ間の入れ子構造を定義するグループ定義手段を有することを特徴とするバルク型時刻認証要求用プログラムある。
【0037】
請求項６に係る発明は、請求項１乃至５に記載のバルク型時刻認証要求用プログラムを記録した、コンピュータ読み取り可能なバルク型時刻認証要求用記録媒体である。
【0038】
ここで、「記録媒体」とは、上記コンピュータで上記各手段を機能させるためのプログラムの読み取りに使用することができればよく、情報を媒体の物理的特性を利用してどのように記録するか等の物理的な記録方法には依存しない。例えば、ＦＤ(Flexible Disk)、ＣＤ－ＲＯＭ（Ｒ，ＲＷ）(Compact Disc Read Only Memory(CD Recordable,CD Rewritable))、ＤＶＤ－ＲＯＭ（ＲＡＭ，Ｒ，ＲＷ）(Digital Versatile Disk Read Only Memory(DVD Random Access Memory,DVD Recordable,DVD Rewritable))、半導体メモリ、ＭＯ(Magneto Optical Disk)、ＭＤ(Mini Disk)、磁気テープ等が該当する（以下、同様）。
【0039】
請求項７に係る発明は、電子フォルダ又は電子ファイルとしての複数のエントリのハッシュ値から新たな１つのハッシュ値を作成し、利用者が前記１つのハッシュ値を時刻認証の証明対象として時刻証明機関へ時刻認証要求を行うことで時刻認証証明書を取得した後に、前記利用者が取得した時刻認証証明書の正当性についての検証を行う検証装置であって、前記複数のエントリ、前記利用者端末で作成されたサマリファイル、及び時刻認証証明書のデータを取得して管理しておく検証用管理手段と、前記エントリに適用するエントリ用ハッシュ関数及びサマリファイルに適用するサマリファイル用ハッシュ関数を管理する関数管理手段と、前記検証用管理手段に管理しておいた時刻認証証明書が改ざんされてなく正しいことを検証する正当性検証手段と、前記検証用管理手段に管理しておいた複数のエントリのうち検証の対象となるものに対して前記関数管理手段で管理しているエントリ用ハッシュ関数を適用して、各エントリのハッシュ値を再計算するハッシュ値再計算手段と、前記検証用管理手段に管理しておいたサマリファイルを構成する要約ファイル表現中の対応する元のエントリのハッシュ値を取得するハッシュ値取得手段と、前記ハッシュ値再計算手段によって再計算した前記エントリのハッシュ値と、前記ハッシュ値取得手段によって取得したハッシュ値とが同じであるか否かを比較するハッシュ値比較手段と、前記検証用管理手段に管理しておいたサマリファイルに対して前記関数管理手段で管理しているサマリファイル用ハッシュ関数を適用して、前記サマリファイルのハッシュ値を再計算するサマリファイルハッシュ値再計算手段と、前記検証用管理手段に管理しておいた時刻認証証明書に含まれる元のサマリファイルのハッシュ値を取得するサマリファイルハッシュ値取得手段と、前記サマリファイルハッシュ値再計算手段で再計算したサマリファイルのハッシュ値と前記サマリファイルハッシュ値取得手段で取得した元のサマリファイルのハッシュ値とが同じであるか否かを比較するサマリファイルハッシュ値比較手段と、を有することを特徴とする検証装置である。
【0040】
請求項８に係る発明は、電子フォルダ又は電子ファイルとしての複数のエントリのハッシュ値から新たな１つのハッシュ値を作成し、利用者が前記１つのハッシュ値を時刻認証の証明対象として時刻証明機関へ時刻認証要求を行うことで時刻認証証明書を取得した後に、前記利用者が取得した時刻認証証明書の正当性についての検証を行う検証装置であっ

て、前記複数のエントリ、前記利用者端末で作成されたサマリファイル、及び時刻認証証明書のデータを取得して管理しておく検証用管理手段と、前記エントリに適用するエントリ用ハッシュ関数及びサマリファイルに適用するサマリファイル用ハッシュ関数を管理する関数管理手段と、前記検証用管理手段に管理しておいた複数のエントリのうち検証の対象となるものに対して前記関数管理手段で管理しているエントリ用ハッシュ関数を適用して、各エントリのハッシュ値を再計算するハッシュ値再計算手段と、前記検証の対象となる各エントリについて、その識別子と上記再計算して得られたエントリのハッシュ値に加えて、上記サマリファイルのハッシュ値を計算するのに必要なデータを、当該のエントリに対する検証用補完データと定義し、さらに、当該のエントリの識別子、そのハッシュ値、そのハッシュ値を計算するためのハッシュ関数の識別子、及び当該のエントリに対する検証用補完データからなるデータを当該エントリの個別化サマリファイルと定義すると、検証の対象となる各エントリに対する個別化サマリファイルから上記サマリファイルのハッシュ値を計算する個別化サマリファイル利用型サマリファイル・ハッシュ値再計算手段と、前記検証用管理手段に管理しておいた時刻認証証明書が改ざんされてなく正しいことを検証する正当性検証手段と、前記検証用管理手段に管理しておいた時刻認証証明書に含まれる元のサマリファイルのハッシュ値を取得するサマリファイルハッシュ値取得手段と、前記個別化サマリファイル利用型サマリファイル・ハッシュ値再計算手段で再計算したサマリファイルのハッシュ値と前記サマリファイルハッシュ値取得手段で取得した元のサマリファイルのハッシュ値とが同じであるか否かを比較するサマリファイルハッシュ値比較手段と、を有することを特徴とする検証装置である。

【0041】

請求項９に係る発明は、電子フォルダ又は電子ファイルとしての複数のエントリのハッシュ値から新たな１つのハッシュ値を作成し、利用者が前記１つのハッシュ値を時刻認証の証明対象として時刻証明機関へ時刻認証要求を行うことで時刻認証証明書を取得した後に、前記利用者が取得した時刻認証証明書の正当性についての検証を行う検証装置を用いた検証方法であって、前記検証装置が、前記複数のエントリ、前記利用者端末で作成されたサマリファイル、及び時刻認証証明書のデータを取得して管理しておく検証用管理手段と、前記エントリに適用するエントリ用ハッシュ関数及びサマリファイルに適用するサマリファイル用ハッシュ関数を管理する関数管理手段とを有した状態において、前記検証装置は、前記検証用管理手段に管理しておいた時刻認証証明書が改ざんされてなく正しいことを検証する正当性検証ステップと、前記検証用管理手段に管理しておいた複数のエントリのうち検証の対象となるものに対して前記関数管理手段で管理しているエントリ用ハッシュ関数を適用して、各エントリのハッシュ値を再計算するハッシュ値再計算ステップと、前記検証用管理手段に管理しておいたサマリファイルを構成する要約ファイル表現中の対応する元のエントリのハッシュ値を取得するハッシュ値取得ステップと、前記ハッシュ値再計算ステップによって再計算した前記エントリのハッシュ値と、前記ハッシュ値取得ステップによって取得したハッシュ値とが同じであるか否かを比較するハッシュ値比較ステップと、前記検証用管理手段に管理しておいたサマリファイルに対して前記関数管理手段で管理しているサマリファイル用ハッシュ関数を適用して、前記サマリファイルのハッシュ値を再計算するサマリファイルハッシュ値再計算ステップと、前記検証用管理手段に管理しておいた時刻認証証明書に含まれる元のサマリファイルのハッシュ値を取得するサマリファイルハッシュ値取得ステップと、前記サマリファイルハッシュ値再計算手段で再計算したサマリファイルのハッシュ値と前記サマリファイルハッシュ値取得ステップで取得した元のサマリファイルのハッシュ値とが同じであるか否かを比較するサマリファイルハッシュ値比較ステップと、を実行することを特徴とする検証方法である。

【0042】

請求項１０に係る発明は、　電子フォルダ又は電子ファイルとしての複数のエントのハッシュ値から新たな１つのハッシュ値を作成し、利用者が前記１つのハッシュ値を時刻認証の証明対象として時刻証明機関へ時刻認証要求を行うことで時刻認証証明書を取得した後に、前記利用者が取得した時刻認証証明書の正当性についての検証を行う検証装置を用

いた検証方法であって、前記検証装置が、前記複数のエントリ、前記利用者端末で作成されたサマリファイル、及び時刻認証証明書のデータを取得して管理しておく検証用管理手段と、前記エントリに適用するエントリ用ハッシュ関数及びサマリファイルに適用するサマリファイル用ハッシュ関数を管理する関数管理手段とを有した状態において、
前記検証装置は、前記検証用管理手段に管理しておいた複数のエントリのうち検証の対象となるものに対して前記関数管理手段で管理しているエントリ用ハッシュ関数を適用して、各エントリのハッシュ値を再計算するハッシュ値再計算ステップと、前記検証の対象となる各エントリについて、その識別子とハッシュ値に加えて、上記サマリファイルのハッシュ値を計算するのに必要なデータを、当該のエントリに対する検証用補完データと定義し、さらに、当該のエントリの識別子、そのハッシュ値、そのハッシュ値を計算するためのハッシュ関数の識別子、及び当該のエントリに対する検証用補完データからなるデータを当該エントリの個別化サマリファイルと定義すると、当該のエントリに対する個別化サマリファイルから上記サマリファイルのハッシュ値を計算する個別化サマリファイル利用型サマリファイル・ハッシュ値再計算ステップと、
前記検証用管理手段に管理しておいた時刻認証証明書が改ざんされてなく正しいことを検証する正当性検証ステップと、前記検証用管理手段に管理しておいた時刻認証証明書に含まれる元のサマリファイルのハッシュ値を取得するサマリファイルハッシュ値取得ステップと、前記個別化サマリファイル利用型サマリファイル・ハッシュ値再計算ステップで再計算したサマリファイルのハッシュ値と前記サマリファイルハッシュ値取得ステップで取得した元のサマリファイルのハッシュ値とが同じであるか否かを比較するサマリファイルハッシュ値比較ステップと、を実行することを特徴とする検証方法である。

【0043】

請求項１１に係る発明は、請求項９に記載の検証装置に、上記各ステップを実行させることを特徴とした検証用プログラムである。

【0044】

請求項１２に係る発明は、請求項１０に記載の検証装置に、上記各ステップを実行させることを特徴とした検証用プログラムである。

【0045】

請求項１３に係る発明は、請求項１1及び請求項１２のいずれか１項に記載のプログラムを記録したことを特徴とした、コンピュータ読み取り可能な検証用記録媒体である。

【発明の効果】

【0046】

以上説明したように本発明によれば、エントリのハッシュ値とエントリの名称を関連付けて作成した要約ファイル表現を合成することで本発明特有のサマリファイルを作成しておくことで、時刻認証証明書が発行された後に、証明対象であるエントリの一部に変更が生じた場合でも、ハッシュ値とエントリの名称との関係が明確なため、変更が生じていない部分は有効であると証明することができるという効果を奏する。さらに、時刻証明の対象となる複数のエントリの中から１つを選んで、そのエントリについて時刻認証証明書が正当であることを第三者において行わせるために、サマリファイルからこの正当性証明に必要な部分のみを抽出した結果である個別化サマリファイルを作成する機能を持つことにより、第三者における正当性検証を、また通信量や処理時間の面で効率的よく、またプライバシに係わる可能性のある情報の開示を最小限にとどめながら可能にするという効果を奏する。

【発明を実施するための最良の形態】

【0047】

〔第１の実施形態〕

以下、図１乃至図９を参照して、本発明の第１の実施形態に係るバルク型時刻認証方法について説明する。尚、図１は、本実施形態における登場主体の関係、通信インフラストラクチャ(Infrastructure)、及び各装置の構成を示した図である。図１に示すように、利用者（甲）は、時刻認証の証明対象としての電子フォルダ又は電子ファイルに対して、ＴＳ

Ａ（乙）に時刻認証証明書の作成を依頼する者であり、パソコン等の利用者端末１からインターネット５を介してＴＳＡ（乙）の時刻認証装置２に時刻認証要求を行うことができる。ＴＳＡ（乙）は、時刻インフラストラクチャ６を介して取得した日本標準時ｔに基づいて、時刻認証証明書 TST1 を作成する者である。但し、日本標準時ｔを取得せずに、独自の時刻源を用いて時刻認証証明書を作成してもよい。

【0048】

尚、以下において、時刻認証の証明対象としての電子フォルダ又は電子ファイルを「エントリ」といい、この場合の電子フォルダには、ある電子フォルダに含まれるもう１つの電子フォルダであるような電子サブフォルダも含まれる。このエントリは、時刻認証の証明対象としてエントリすることに由来している。また、電子ファイルを「ファイル」として表し、電子サブフォルダを「サブフォルダ」として表し、電子ファイルを「ファイル」として表す。ファイル毎に各デジタルデータが格納されている。

【0049】

ファイルとフォルダのうちのどちらを示しても良い場合、又は両方を示す場合の概念の総称を「エントリｅ」として表す。また、各エントリｅをそれぞれ示す場合には、エントリｅ１，ｅ２，・・・，ｅｎ（ｎは正数）として表す。

【0050】

特に、エントリｅがフォルダの場合には総称として「エントリeF」と表し、エントリｅがサブフォルダの場合には総称として「エントリes」と表し、エントリｅがファイルの場合には総称として「エントリef」と表す。

この場合、サブフォルダに係るエントリesをそれぞれ示す場合には、エントリes１，es２，・・・，esｎ（ｎは正数）として表し、ファイルに係るエントリefをそれぞれ示す場合には、エントリef1，ef2，・・・，efｎ（ｎは正数）として表す。

【0051】

係争相手（丙）は、利用者（甲）が自己のデジタルデータの作成日時等に関して争う当事者の一方であり、パソコン等の係争者端末３からインターネット５を介して利用者端末１又は時刻認証装置２に対し、デジタルデータの送信要求を行うことができる。

【0052】

ＴＴＰ（丁）は、信頼される第三者機関として、中立性と客観性をもち、利用者（甲）と係争相手（丙）の紛争解決に役立てるために時刻認証証明書の正当性を検証する機関であり、検証装置４によりインターネット５を介して利用者端末１からの時刻証明検証要求を受け付けて、時刻認証証明書の正当性の検証を行うと共に、インターネット５を介して検証結果を利用者端末１又は係争者端末３に送信することができる。尚、利用者（甲）は利用者端末１を利用せずに、ＣＤ－Ｒ等の記録媒体に時刻証明検証要求に必要なデータを記録しておき、ＴＴＰ（丁）に持参や郵送等により、上記時刻証明検証要求に必要なデータを渡すようにしてもよい。

【0053】

次に、利用者端末１、時刻認証装置２、及び検証装置４について説明する。尚、利用者端末１には、この利用者端末１を図１に示す各手段として機能させるための時刻認証要求用プログラム(p1)が記憶されている。また、時刻認証装置２には、この時刻認証装置２を図１に示す各手段として機能させるための時刻認証用プログラム(p2)が記憶されている。更に、検証装置４には、この検証装置４を図１に示す各手段として機能させるための正当性検証用プログラム(p4)が記憶されている。

【0054】

また、コンピュータとしての利用者端末１は、時刻認証要求用プログラム(p1)を実行することによって、図１に示すように、エントリ指定手段１０、サマリファイル作成手段１１、サマリファイル記憶手段１２、サマリファイルハッシュ値作成手段１４、及び、送受信手段１５を有するに至っている。

【0055】

更に、サマリファイル作成手段１１は、ハッシュ値作成手段１１ａ、ハッシュ値結合手

段１１ｂ、統合ハッシュ値作成手段１１ｃ、要約ファイル表現作成手段１１ｄ、及び要約ファイル表現合成手段１１ｅを有している。

【0056】

また、利用者端末１における不図示のＨＤ(Hard Disk)には、関数管理手段１８、及び、ファイル管理手段１９が構築されている。

【0057】

また、関数管理手段１８には、ファイルに係るエントリefに適用してハッシュ値$h1_{ef}$(ef)を計算するためのファイル用ハッシュ関数$h1_{ef}$(以下、「ハッシュ関数$h1_{ef}$」という)、フォルダに係るエントリeFの統合ハッシュ値を計算するための統合用ハッシュ関数$h2_{eF}$（以下、「ハッシュ関数 $h2_{eF}$」という）、サブフォルダに係るエントリesの統合ハッシュ値を計算するための統合用ハッシュ関数$h2_{es}$（以下、「ハッシュ関数 $h2_{es}$」という）、及び、サマリファイルＳＦに適用してサマリファイルのハッシュ値$h3_{eF}$(ＳＦ)を計算するためのサマリファイル用ハッシュ関数$h3_{eF}$（以下、「ハッシュ関数$h3_{eF}$」という）が記憶されて管理されている。尚、ファイル用ハッシュ関数 $h1_{ef}$、統合用ハッシュ関数 $h2_{eF}$ 及び、統合用ハッシュ関数 $h2_{es}$ は、エントリ用ハッシュ関数の一例である。

【0058】

また、ハッシュ関数 $h1_{ef}$, $h2_{eF}$, $h2_{es}$, $h3_{eF}$ として、MD5、SHA-1等がある。また、ハッシュ関数$h1_{ef}$, $h2_{eF}$, $h2_{es}$, $h3_{eF}$ は、全て同じものであってもよく、別のものであってもよいし、これらのうちのいくつかが同じものであってもよい。これらのうちで、$h3_{eF}$ はサマリファイルを入力してそのハッシュ値を出力する関数であるが、サマリファイルが曖昧性なく構造を表現するマークアップ言語などで記述されている場合には、$h3_{eF}$ はマリファイルをバイト列等のデジタルデータとして扱い、このようなデジタルデータに適用する上記MD5, SHA-1 等のハッシュ関数を適用するものであってもよいし、またサマリファイルで表現された構造を抽出しその構造に基づく何らかの変換を行った後に、この変換結果をバイト列等のデジタルデータとして扱い、このようなデジタルデータに適用する上記MD5, SHA-1 等のハッシュ関数を適用するものであってもよい。このような変換の例としては、サマリファイル内の一個以上の特定のフィールドを抽出する方式がある。尚ハッシュ値は、別名「ダイジェスト(digest)」ともいう。図６では、<digest>として表している。

【0059】

更に、関数管理手段１８には、サマリファイルＳＦを構成する要約ファイル表現ＳＦＲのうちで、エントリeF, esに基づいた要約ファイル表現 SFR(eF), SFR(es) を作成するためのフォルダ用要約ファイル表現作成関数 FolderSfRepが記憶されて管理されている。

【0060】

このフォルダ用要約ファイル表現作成関数 FolderSfRep は、まず、サマリファイルＳＦを構成する要約ファイル表現ＳＦＲのうちで、ファイルに係るエントリefに基づいた要約ファイル表現 SFR(ef) を作成するためのファイル用要約ファイル表現作成関数 FileSfRep を、各エントリefに適用する機能を有すると共に、これによって作成した複数の要約ファイル表現 SFR(ef) を合成して、結果的にサマリファイルＳＦを構成する要約ファイル表現ＳＦＲを完成させるための要約ファイル表現合成関数 ComposeSFR を適用する機能を有している。

【0061】

また、サブフォルダが存在する場合には、サブフォルダに係るエントリesに基づいた要約ファイル表現 SFR(es) を作成のため、上記フォルダ用要約ファイル表現作成関数 FolderSfRep をサブフォルダ用に再帰的に適用する。この場合、最終的にエントリeFに基づいた要約ファイル表現 SFR(eF) を作成する際に、要約ファイル表現合成関数 ComposeSFRを適用する対象は、(1)フォルダ用要約ファイル表現合成関数 FolderSfRepによって作成した要約ファイル表現SFR(es) か、又は、(2)この要約ファイル表現 SFR(es) 及びファイル用要約ファイル表現作成関数 FileSfRepによって作成した要約ファイル表現 SFR(ef) の組み合わせである。

【0062】
次に、エントリ指定手段１０は、複数のファイル及び複数のフォルダに係るデジタルデータＧのうち、利用者（甲）によって入力されたエントリｅの識別子Id(e) 又はパス名epnに基づいて、時刻認証の証明対象となるエントリ（ｇ１，ｇ２，・・・，ｇｎ）を指定する機能を有する。この複数のファイル及び複数のフォルダは、利用者端末１のＨＤに記憶されている場合や、外部のＣＤ－Ｒ等の記録媒体に記録されている場合や、インターネット等の通信ネットワークを介して送られてくる場合等がある。
【0063】
また、ハッシュ値作成手段１１ａは、関数管理手段１８からハッシュ関数$h1_{ef}$を読み出して、上記エントリ指定手段１０によって指定したファイルに係る各エントリefに対してハッシュ関数 $h1_{ef}$ を適用することで、これらエントリef毎にハッシュ値 $h1_{ef}$(ef) を作成する機能を有する。また、サブフォルダに係るエントリesに格納されているエントリefについても、ハッシュ値作成手段１１ａによってハッシュ値 $h1_{ef}$(ef) を作成する。
【0064】
また、ハッシュ値結合手段１１ｂは、エントリの識別子 Id(ef)，ハッシュ値作成手段１１ａで作成した各ハッシュ値 $h1_{ef}$(ef) と、MD5等のハッシュ・アルゴリズム(algorithm)の名称等のそれぞれの付加情報α１，α２，、、、，αｎ（ｎは正数。以下、これらの総称を「α」と表す。）を連接して、予め定められた順序付けの規則に従って、以下のような系列に並べ替える機能を有する。
【0065】
Id(ef1)‖$h1_{ef1}$(ef1)‖α１、
Id(ef2)‖$h1_{ef2}$(ef2)‖α２、
・・・
Id(efn)‖$h1_{efn}$(efn)‖αｎ
尚、順序付けの規則としては、エントリefの識別子 Id(ef) に基づく辞書式順序に従って定めることが挙げられる。また、付加情報αは空でもよい。
【0066】
更に、ハッシュ値結合手段１１ｂは、上記連接した系列を結合して、１つのデジタルデータを生成する機能も有する。
【0067】
また、統合ハッシュ値作成手段１１ｃは、関数管理手段１８からハッシュ関数$h2_{eF}$を読み出して、上記ハッシュ値結合手段１１ｂによって結合した値に適用することで、上記複数のエントリefが格納されているフォルダに係るエントリeFの統合ハッシュ値を作成する機能を有する。この作成した統合ハッシュ値は、以下に示す通りである。
【0068】
$h2_{eF}$(Id(ef1)‖$h1_{ef1}$(ef1)‖α１‖Id(ef2)‖$h1_{ef2}$(ef2)‖α２‖
…‖Id(efn)‖$h1_{efn}$(efn)‖αｎ)
尚、フォルダに係るエントリeFにサブフォルダに係るエントリesが格納されている場合には、上記フォルダの概念にサブフォルダを含めて再帰的に上記の手順を適用することで、サブフォルダに係るエントリesの統合ハッシュ値を作成し、その後に、フォルダに係るエントリeFの統合ハッシュ値を作成する。
【0069】
次に、要約ファイル表現作成手段１１ｄは、関数管理手段１８からファイル用要約ファイル表現作成関数FileSfRepを読み出して上記エントリef毎に適用することで、少なくとも（１）上記入力された証明対象のフォルダ（エントリ）のパス名epnに基づく各ファイル（エントリ）の識別子及び（２）上記ハッシュ値作成手段１１ａによって作成したハッシュ値 $h1_{ef}$(ef) を、所定の規則に従い曖昧性なく関連づけて記述する記述言語（ＸＭＬ(Extensible Markup Language)等のマークアップ言語等）により記述した要約ファイル表現 SFR(ef) を作成する機能を有する。また、この要約ファイル表現 SFR(ef) は、上記エントリef 毎に作成される。

【0070】
ここで、ファイル用要約ファイル表現作成関数 FileSfRep は、図３に示すように、引数となるファイルの識別子と、HashRep(e)とによって構成された値を求めるために使用される。また、HashRep(e)は、図４に示すように、ハッシュデータ<hashdata>で表現されている。このハッシュデータには、ハッシュ・アルゴリズム名<algorithm>及びハッシュ値（ダイジェスト）<digest>で表現されている。尚、図４は、SHA-1やMD5等のハッシュアルゴリズムのように、各エントリｅ専用のハッシュキー(hashkey)を使わない場合を示している。
【0071】
また、各エントリｅ専用のハッシュキーを使う場合については、図５に示している。この場合には、ハッシュデータに、更に、ハッシュ・キーの値<hashkey>の表現も含まれる。この場合のハッシュアルゴリズムは、HMAC-SHA-1やHMAC-MD5等である。
【0072】
また、要約ファイル表現合成手段１１ｅは、関数管理手段１８から要約ファイル表現合成関数 ComposeSFR を読み出して、上記要約ファイル表現作成手段１１ｄによって作成した各ファイル用要約ファイル表現 SFR(ef) にまとめて適用することで、複数のファイル用要約ファイル表現 SFR(ef) を合成する。
【0073】
更に、要約ファイル表現作成手段１１ｄは、上記合成した値に対して、少なくとも（１）上記利用者（甲）によって入力されたエントリｅのパス名epnに基づくフォルダの識別子及び（２）上記統合ハッシュ値作成手段１１ｃによって作成した統合ハッシュ値を所定の規則に従い曖昧性なく関連付けて記述する記述言語（ＸＭＬ等のマークアップ言語等）により表現したフォルダ用要約ファイル表現 SFR(eF) を作成する機能を有する。尚、エントリeFが木構造を有する場合には、この木構造を反映したフォルダ用要約ファイル表現 SFR(eF) としてもよく、上記エントリeFの木構造とは全く異なる新たな木構造のフォルダ用要約ファイル表現SFR(eF) としてもよい。
【0074】
尚、フォルダに係るエントリeFに、サブフォルダに係るエントリesが格納されている場合には、上記要約ファイル表現作成手段１１ｄは、サマリファイルＳＦ内のフォルダ用要約ファイル表現 SFR(eF) の概念に、上記サブフォルダに係るエントリesに基づくサブフォルダ用要約ファイル表現 SFR(es) を含めて、再帰的手順でサブフォルダ用要約ファイル表現 SFR(es) 及びフォルダ用要約ファイル表現SFR(eF) を作成する。この場合には、上記要約ファイル表現合成手段１１ｅは、上記ファイル用要約ファイル表現 SFR(ef) と同様に上記作成したサブフォルダ用要約ファイル表現 SFR(es) を扱い再帰的に要約ファイル表現合成関数 ComposeSFR を適用する。
【0075】
これにより、要約ファイル表現作成手段１１ｄによって、サブフォルダ用要約ファイル表現SFR(es) を作成した場合には、要約ファイル表現合成手段１１ｅによって合成する対象は、ファイル用要約ファイル表現 SFR(ef) だけでなく、上記一旦作成したサブフォルダ用要約ファイル表現 SFR(es) も含まれることになる。
【0076】
この場合、要約ファイル表現合成手段１１ｅによる合成は、予め定められた順序付けの規則に従って、以下のような系列に並べ替えられる。
【0077】
SFR(ｅ１)、
SFR(ｅ２)、
　・・・
SFR(ｅｎ)
尚、順序付けの規則の例としては、エントリｅの識別子に基づく辞書式順序に従って定めることが挙げられる。

【0078】
また、要約ファイル表現合成手段１１ｅは、以下に示す方法によって、具体的に要約ファイル表現 SFR(ｅ）を結合することで、木構造を表現したサマリファイルＳＦを作成する。
【0079】
まず、要約ファイル表現合成手段１１ｅは、上記利用者（甲）によって入力されたフォルダ（エントリｅ）のパス名epnと、上記要約ファイル表現作成手段１１ｄによって作成して並べ替えた各要約ファイル表現 SFR(ｅ）に対して要約ファイル表現合成関数 ComposeSFRを適用し、エントリｅに基づく要約ファイル表現 SFR(ｅ）を作成する。この要約ファイル表現合成関数 ComposeSFRは、予め定められた関数であり、例えばエントリｅに基づく要約ファイル表現 SFR(ｅ）を入力した場合に、ＸＭＬ形式で図２に示すように表現することができる関数である。但し、図２において、下線を引いた式（例えば、SFR(ｅ１））は、その式そのものではなく、つまり、文字列としての SFR(ｅ１）そのものではなく、その式が表す文字列を、その場所に代入するという事を示している。
【0080】
図２では、上記入力されたフォルダのパス名epnに基づいたフォルダネーム<folder name>、上記図４又は図５で示した HashRep(e)、及び、上記要約ファイル表現 SFR(ｅ）によって表現されている。
【0081】
以上のように、サマリファイル作成手段１１によって作成されたサマリファイルＳＦの一例であるサマリファイルＳＦ１を図６に示す。図６に示すように、サマリファイルＳＦ１は、ユーザネーム<username>等の管理上必要な情報としての附加要素と、フォルダに係るエントリeF1に基づいた要約ファイル表現SFR(eF1）によって構成されている。この要約ファイル表現SFR(eF1）は、元のデータがフォルダであるため、図２に示す表現手法を採用し、HashRep(eF1)、エントリes１に基づいた要約ファイル表現SFR(es1）、及び、エントリef2に基づいた要約ファイル表現SFR(ef2）によって構成されている。
【0082】
更に、要約ファイル表現SFR(es１）は、エントリes1が（サブ）フォルダであるため、図２に示す表現手法を採用し、HashRep(es1)及びエントリef2に基づいた要約ファイル表現SFR(ef2）によって構成されている。また、要約ファイル表現SFR(ef3）は、エントリef3がファイルであるため、図３に示す表現手法を採用し、HashRep(ef3)によって構成されている。
【0083】
一方、要約ファイル表現SFR(ef2）は、エントリ ef2がファイルであるため、図３に示す表現手法を採用し、HashRep(ef2)によって構成されている。
【0084】
このように、サマリファイルＳＦは、以下に示す複数の特徴を有する。
【0085】
〔１〕ＸＭＬ等の特定の記述言語により記述しているため、この記述から曖昧性なく各構成要素とそれらの関係を抽出することができる。
【0086】
〔２〕エントリの名称とエントリのハッシュデータが関連付けられて、エントリｅのハッシュデータ<hashdata>が表示されている。
【0087】
〔３〕ハッシュデータには、使用したハッシュアルゴリズム<algorithm>、計算したハッシュ値<digest>、場合によってはハッシュキー<hashkey>も示されている。
【0088】
〔４〕木構造をテキストファイルとして表現しており、木構造を作成すること自体が、サマリファイルを作成することになる。
【0089】

次に、図１に示すサマリファイル記憶手段１２は、サマリファイル作成手段１１によって作成したサマリファイルＳＦをファイル管理手段１９に記憶して管理する機能を有する。そして、係争相手（丙）と時刻認証証明書の正当性について争いが生じた場合に、ファイル管理手段１９に管理しておいたサマリファイルＳＦを読み出して、正当性の検証に役立てる。

【0090】

また、サマリファイルハッシュ値作成手段１４は、関数管理手段１８からハッシュ関数$h3_{eF}$を読み出して、ファイル管理手段１９に記憶すべき完成されたサマリファイルＳＦに適用することで、サマリファイルのハッシュ値$h3_{eF}$（ＳＦ）を作成する。尚、説明の便宜上、以下、サマリファイルのハッシュ値$h3_{eF}$（ＳＦ）を「Ｄ」で表現する。

【0091】

また、送受信手段１５は、インターネット５を介して他の装置と情報（データ）の送受信を行うための機能を有する。

【0092】

続いて、図１に示す時刻認証装置２について説明する。時刻認証装置２は、時刻認証用プログラム(p2)を実行することによって、図１に示すように、送受信手段２１、時刻取得手段２２、時刻認証手段２３、及び、デジタル署名作成手段２４を有するに至っている。

【0093】

送受信手段２１は、インターネット５を介して他の装置と情報（データ）の送受信を行うための機能を有する。ここでは、インターネット５を介して利用者端末１から送られてきたハッシュ値Ｄを受信する。また、時刻取得手段２２は、送受信手段２１でハッシュ値Ｄを受信した時に合わせて、時刻インフラストラクチャ６から日本標準時ｔを取得する機能を有する。尚、日本標準時ｔを取得せずに、独自の時刻源を用いてもよい。

【0094】

また、デジタル署名作成手段２３は、図９に示すように、送受信手段２１で受信したハッシュ値Ｄのデータに、時刻取得手段２２で取得した時刻ｔ、および所定の附加情報を結合して、署名対象デジタルデータTSTI(D, t)とし、この結合した全体に対してデジタル署名 sig(SK, TSTI(D, t)) を作成する機能を有する。このデジタル署名 sig(SK, TSTI(D, t))は、ＴＳＡ（乙）の署名鍵（秘密鍵）SKによるデジタルデータ TSTI(D, t)に対するデジタル署名を示したものである。

【0095】

また、時刻認証手段２４は、上記結合したハッシュ値Ｄと時刻ｔ、及び所定の附加情報を結合して構成した 署名対象デジタルデータ TSTI(D, t)、及び、上記作成されたデジタル署名 sig(SK, TSTI(d, t)) を含む時刻認証証明書 TST(sig(SK, TSTI(D, t)), TSTI(D, t))を作成する機能を有する。この時刻認証証明書 TST(sig(SK, TSTI(D, t)), TSTI(D, t)) は、署名付きデジタルデータとしての時刻認証証明書を示したものである。

【0096】

続いて、図１に示す検証装置４について説明する。

【0097】

検証装置４は、検証用プログラム(p4)を実行することによって、図１に示すように、送受信手段４１、及び、検証手段４２を有するに至っている。更に、検証手段４２は、正当性検証手段４２ａ、ハッシュ値再計算手段４２ｂ１、ハッシュ値取得手段４２ｂ２、ハッシュ値比較手段４２ｂ３、サマリファイルハッシュ値再計算手段４２ｃ１、サマリファイルハッシュ値取得手段４２ｃ２、及び、サマリフィルハッシュ値比較手段４２ｃ３を有している。また、コンピュータとしての検証装置４における不図示のＨＤには、検証用管理手段４８、及び、関数管理手段４９が構築されている。この関数管理手段４９には、上記ハッシュ関数$h1_{ef}$, $h2_{eF}$, $h3_{eF}$ が既に管理されている。尚、署名鍵SKと対になる公開鍵PKは、何らかの手段でＴＴＰ（丁）に知られている。

【0098】

検証装置４のうち、送受信手段４１は、インターネット５を介して他の装置と情報（デ

ータ）の送受信を行い、受信した情報を検証用管理手段４８に記憶して管理する機能を有する。例えば、利用者端末１から時刻証明検証要求時に送られてきた、単一若しくは複数のエントリｅ、このエントリｅに基づくサマリファイルＳＦ、及び、時刻認証証明書 TST(sig(SK, TSTI(D, t)), TSTI(D, t))を受信し、検証用記憶手段４９に記憶しておく。

【0099】

また、検証手段４２は、以下に示す機能により、時刻認証証明書の正当性の検証を行う。

【0100】

正当性検証手段４２ａは、検証用管理手段４８に管理しておいた時刻認証証明書 TST(sig(SK, TSTI(D, t)), TSTI(D, t)) が改ざんされてなく正しいことを検証する機能を有する。例えば、検証用管理手段４８に管理しておいた時刻認証証明書 TST(sig(SK, TSTI(D, t)), TSTI(D, t)) に含まれるデジタル署名 sig(SK, TSTI(D, t)) が、上記時刻認証証明書 TST(sig(SK, TSTI(D, t)), TSTI(D, t)) に含まれる署名対象デジタルデータ TSTI(D, t)に対するデジタル署名であることを検証する。

【0101】

ハッシュ値再計算手段４２ｂ１は、ＴＴＰ（丁）によって入力された検証対象のエントリｅのパス名epnに基づいて、検証用管理手段４８から検証対象のエントリｅを読み出すと共に、関数管理手段４９からハッシュ関数$h1_{ef}$, $h2_{es}$を読み出して上記エントリｅ毎に適用することで、各ハッシュ値を再計算する機能を有する。この再計算方法は、上記利用者端末１のハッシュ作成手段１１ａ又は統合ハッシュ値作成手段１１ｃで各ハッシュ値を作成した手順と同じ手順で行う。

【0102】

ハッシュ値取得手段４２ｂ２は、検証用管理手段４８に管理しておいたサマリファイルＳＦを構成する要約ファイル表現SFR(ｅ)中の対応する元のエントリのハッシュ値D1を取得する機能を有する。

【0103】

ハッシュ値比較手段４２ｂ３は、ハッシュ値再計算手段４２ｂ１によって再計算したエントリｅのハッシュ値 D1'と、ハッシュ値取得手段４２ｂ２によって取得した元のエントリのハッシュ値D1とが同じ（D1'＝D1）であるか否かを比較する機能を有する。

【0104】

サマリファイルハッシュ値再計算手段４２ｃ１は、検証用管理手段４８に管理しておいたサマリファイルＳＦに対して関数管理手段４９で管理しているサマリファイル用ハッシュ関数$h3_{eF}$を適用して、サマリファイルＳＦのハッシュ値$h3_{eF}$（ＳＦ）を再計算する機能を有する。

【0105】

尚、以下、このサマリファイルハッシュ値再計算手段４２ｃ１によって再計算して求めたハッシュ値を「D'」で表す。この再計算方法は、上記利用者端末１のサマリファイルハッシュ値作成手段１４によって、サマリファイルＳＦのハッシュ値Ｄを作成した手順と同じ手順で行う。

【0106】

サマリファイルハッシュ値取得手段４２ｃ２は、前記検証用管理手段４８に管理しておいた時刻認証証明書 TST(sig(SK, TSTI(D, t)), TSTI(D, t)) の中の署名対象デジタルデータTSTI(D, t)に含まれる元のサマリファイルのハッシュ値Ｄを取得する機能を有する。

【0107】

サマリファイルハッシュ値比較手段４２ｃ３は、サマリファイルハッシュ値再計算手段４２ｃ１で再計算したサマリファイルＳＦのハッシュ値D'と、正当性検証手段４２ａで取得した元のサマリファイルＳＦのハッシュ値Ｄとが同じである（D'＝D）か否かを比較する機能を有する。

【0108】

以上により、検証装置４としては、D1'＝D1、及び、D'＝Dが成り立てば、少なくとも、

ハッシュ値D1’に係るエントリが正当であり時刻印を押された時点以降において変更されていないと判断できるため、ハッシュ値 D1’に係るエントリはの検出結果として「正当（ＯＫ）」を出力し、D’＝Dが成り立たなければ、検出結果として「不当（ＮＧ）」を出力する。

【0109】

続いて、主に図８及び図９を用いて、本実施形態の動作について説明する。尚、ここでは、図６に示すサマリファイルＳＦ１を作成して、ＴＳＡ（乙）に時刻認証要求する場合について説明する。

【0110】

まず、利用者（甲）が、利用者端末１に証明対象のフォルダに係るエントリeF1のパス名epnを入力すると、エントリ指定手段１０では、パス名epnに基づいて、フォルダ管理手段１７からエントリeF1を読み出す。このエントリeF1は、図７に示すような木構造により構成されている。即ち、エントリeF1には、サブフォルダであるエントリes1とファイルであるエントリef2が含まれており、更に、エントリes1には、ファイルであるエントリef3が含まれている。このように、１つのエントリeF1に含まれるエントリes1とエントリef3の間に包含関係がある場合、そのエントリef3からエントリes1への有向枝があることにすれば、１つのエントリeF1に含まれるエントリes1とエントリef3の全体から木構造を構成することができる。

【0111】

そして、サマリファイル作成手段１１は、図８に示す手順で、サマリファイルＳＦ１を作成する。

【0112】

まず、図７に示す最上位階層であるエントリeF1に基づく要約ファイル表現SFR(eF1）を作成するため、要約ファイル表現作成手段１１ｄによって、関数管理手段１８からフォルダ用要約ファイル表現作成関数 FolderSfRepを読み出す（ステップＳ１）。

【0113】

しかし、エントリeF1には、その下の階層として、サブフォルダに係るエントリes1とファイルに係るエントリef2が格納されているため、エントリes1に基づいた要約ファイル表現SFR(es1）と、エントリef2に基づいた要約ファイル表現SFR(ef2）を作成する必要がある。そのため、更に、要約ファイル表現作成手段１１ｄによって、関数管理手段１８からフォルダ用要約ファイル表現作成関数FolderSfRepを読み出すと共に、要約ファイル表現合成手段１１ｅによって、関数管理手段１８からファイル用要約ファイル表現作成関数FileSfRep読み出す（ステップＳ２）。

【0114】

しかし、エントリes1には、その更に下の階層として、ファイルに係るエントリef3が格納されているため、エントリef3に基づいた構成要素SFR(ef3）を作成する必要がる。よって、更にまた、要約ファイル表現合成手段１１ｅによって、関数管理手段１９からファイル用要約ファイル表現作成関数FileSfRepを読み出す（ステップＳ３）。

【0115】

次に、要約ファイル表現SFR(ef3）を作成するため、まず、ハッシュ値作成手段１１ａによって、関数管理手段１８からハッシュ関数$h1_{ef3}$を読み出して（ステップＳ４ａ）、エントリef3に適用することでハッシュ値$h1_{ef3}$(ef3）を作成する（ステップＳ４ｂ）。更に、要約ファイル表現合成手段１１ｅによって、上記ハッシュ値$h1_{ef3}$ (ef3）を用いると共に、上記ステップＳ３で読み出したファイル用要約ファイル表現作成関数FileSfRepをエントリef3に適用して、要約ファイル表現SFR(ef3）を作成する（ステップＳ４ｃ）。

【0116】

これにより、図７に示すエントリes1に基づく要約ファイル表現SFR(es1）の構成要素が確定したため、次に、要約ファイル表現SFR(es1）を作成する。そのため、まず、ハッシュ値結合手段１１ｂによりハッシュ値$h1_{ef3}$(ef3）と付加情報α３との結合を行うと共に、統合ハッシュ値作成手段１１ｃによって、関数管理手段１８からハッシュ関数$h2_{es1}$を

読み出して（ステップＳ５ａ）、結合した結果 $h1_{ef3}(ef3) \| \alpha 3$ に適用することで$h2_{es1}$による es1 の統合ハッシュ値を作成する（ステップＳ５ｂ）。更に、要約ファイル表現合成手段１１ｅによって、上記$h2_{es1}$による es1 の統合ハッシュ値を用いると共に、上記ステップＳ２で読み出したフォルダ用要約ファイル表現作成関数FolderSfRepをエントリes1に適用して、要約ファイル表現SFR(es1）を作成する（ステップＳ５ｃ）。

【0117】

また、図７に示すエントリeF1に基づく要約ファイル表現SFR(eF1）の構成要素を確定させるために、更に、要約ファイル表現SFR(ef1）を作成する。そのために、まず、ハッシュ値作成手段１１ｄによって、関数管理手段１８からハッシュ関数$h1_{ef2}$を読み出して（ステップＳ６ａ）、エントリef2に適用することでハッシュ値$h1_{ef2}$(ef2）を作成する（ステップＳ６ｂ）。更に、要約ファイル表現合成手段１１ｄａによって、上記ハッシュ値 $h1_{ef2}$(ef2）を用いると共に、上記ステップＳ２で読み出したファイル用要約ファイル表現作成関数FileSfRepをエントリef2に適用して、要約ファイル表現SFR(ef2）を作成する（ステップＳ６ｃ）。

【0118】

これにより、図７に示すエントリeF1に基づく要約ファイル表現SFR(eF1）の構成要素が確定したため、ようやく要約ファイル表現SFR(eF1）を作成することができる。そこで、まず、ハッシュ値結合手段１１ｂにより$h2_{es1}$によるes1の統合ハッシュ値と付加情報$\alpha 1$との結合を行うと共に、更に続けて、ハッシュ値$h1_{ef2}$(ef2）と付加情報$\alpha 2$との結合を行う。そして、統合ハッシュ値作成手段１１ｃによって、関数管理手段１８から統合用ハッシュ関数$h2_{eF1}$を読み出して（ステップＳ７ａ）、上記のように結合した結果，即ち、$h2_{es1}$によるes1の統合ハッシュ値、$\alpha 1$，$h1_{ef2}$(ef2），及び$\alpha 2$の連接に適用することで$h2_{eF1}$による eF1の統合ハッシュ値を作成する（ステップＳ７ｂ）。更に、要約ファイル表現合成手段１１ｅによって、上記$h2_{eF1}$による eF1の統合ハッシュ値を用いると共に、上記ステップＳ１で読み出したフォルダ用要約ファイル表現作成関数FolderSfRepをエントリeF1に適用して、要約ファイル表現SFR(eF1）を作成する（ステップＳ７ｃ）。

【0119】

そして最後に、サマリファイル作成手段１１によって、管理用等の附加要素を付加することで（ステップＳ８）、図７に示すようなサマリファイルＳＦ１が完成することになる。

【0120】

次に、サマリファイル記憶手段１２によって、上記作成したサマリファイルＳＦ１をファイル管理手段１９に記憶して管理しておく。また、サマリファイルハッシュ値作成手段１４によってハッシュ値Ｄを作成する。そして、送受信手段１５によって、図９に示すように、ＴＳＡ（乙）の時刻認証装置２へハッシュ値Ｄを送信することで、利用者（甲）がＴＳＡ（乙）に時刻認証要求を行う（ステップＳ１１）。

【0121】

次に、ＴＳＡ（乙）の時刻認証装置２では、時刻取得手段２２によって取得した時間ｔに基づいてデジタル署名作成手段２３でデジタル署名 sig2を作成し、時刻認証手段２４により、時刻認証証明書 TST2を作成する。そして、図９に示すように、ＴＳＡ（乙）は、時刻認証装置２から利用者（甲）の利用者端末１へ時刻認証証明書 TST2を送信することで、時刻証明応答を行う。（ステップＳ１２）。

【0122】

次に、利用者（甲）が、フォルダに係るエントリeF1の時刻証明を係争相手（丙）に証明する必要が出てきた場合には、図９に示すように、利用者（甲）は、利用者端末１からインターネット５を介してＴＴＰ（丁）の検証装置４に、証明対象であるエントリｅ、このフォルダｅに基づくサマリファイルＳＦ、及び、時刻認証証明書 TST(sig(SK, TSTI(D, t)), TSTI(D, t))を送信することで、時刻証明検証要求を行う（ステップＳ１３）。

【0123】

次に、検証装置４では、送受信手段４１により受信したエントリｅ、サマリファイルＳ

Ｆ、及び、時刻認証証明書 TST(sig(SK, TSTI(D, t)), TSTI(D, t))を検証用管理手段４８に記憶して管理し、この検証用管理手段４８及び関数管理手段４９と検証手段４２との協働によって、時刻認証証明書 TST(sig(SK, TSTI(D, t)), TSTI(D, t)) の正当性の検証を行う。そして、この検証結果は、図９に示すように、時刻証明検証応答として、送受信手段４１により、インターネット５を介して利用者端末１や係争者端末３に送信される（ステップＳ１４）。

【0124】

以上説明したように本実施形態によれば、エントリｅの名称や当該エントリｅのハッシュ値のデータをＸＭＬ等のように、木構造を所定の規則に従って曖昧さなく表現するような特定の記述言語（必要ならその規則に従って人間がこの木構造を理解することができるような記述言語）により表現することで複数の要約ファイル表現ＳＦＲを作成し、更に、これら複数の要約ファイル表現ＳＦＲから成る木構造を構成することで、最終的にサマリファイルＳＦを作成している。このため、サマリファイルＳＦをファイル管理手段１９に記憶して管理しておけば、エントリeF1のうちの一部のエントリ（例えば、エントリef3）に変更が生じた場合でも、人間がサマリファイルＳＦの要約ファイル表現ＳＦＲを見て、又は、コンピュータによる自動解析により、変更された部分だけは無効で、それ以外の部分（例えば、エントリef2）は有効であると検証することができる。

【0125】

また、要約ファイル表現作成手段１１ｄによって、サマリファイルＳＦを、エントリｅの名称と当該エントリｅのハッシュ値 Dの対応付けをして表現しているため、ハッシュ値がどのエントリｅ（デジタルデータｇ）のものかを認識するために、１つ１つハッシュ値の再計算をしなければならないという手間を省略することができる。

【0126】

〔第２の実施形態〕

以下、図１０及び図１１を参照して、本発明の第２の実施形態に係るバルク型時刻認証方法について説明する。尚、本実施形態では、サブフォルダのないフラットな木構造によって構成されたフォルダに係るエントリef2を証明対象にした場合について説明している。よって、利用者端末１の構成要素は、上記第１の実施形態の構成要素と同じであるため、その説明を省略する。

【0127】

図１０に示すサマリファイルＳＦ２は、図１１に示すような木構造の構成になっている。図１１には、サブフォルダのないフラットな木構造によって構成されたエントリef2を示している。図１１に示すように、エントリef2には、サブフォルダを含まずに、ファイルに係るエントリef11，ef12が格納されている。

【0128】

また、サマリファイルＳＦ２の構成は、図１０に示すように、ユーザネーム<username>等の管理用等の附加要素と、エントリeF2に基づいた要約ファイル表現SFR(eF2) によって構成されている。この要約ファイル表現SFR(eF2) は、元のデータがフォルダであるため、図２に示す表現手法を採用し、HashRep(eF2)、エントリef11に基づいた要約ファイル表現SFR(ef11)、及び、エントリef12に基づいた要約ファイル表現SFR(ef12) によって構成されている。

【0129】

更に、要約ファイル表現SFR(ef11) は、エントリef11がファイルであるため、図３に示す表現手法を採用し、HashRep(ef11)によって構成されている。同様に、要約ファイル表現SFR(ef12) は、エントリef12がファイルであるため、図３に示す表現手法を採用し、HashRep(ef12)によって構成されている。

【0130】

以上説明したように本実施形態によれば、上記第１の実施形態と同様の効果を奏することができる。

【0131】

〔第３の実施形態〕

以下、図１２乃至図１５を参照して、本発明の第３の実施形態に係るバルク型時刻認証方法について説明する。

【0132】

図１２は、ＬＡＮ(Local Area Network)やインターネット等の通信ネットワークで接続された２つのパソコン（パーソナル・コンピュータ）P1，P2と、これらパソコンP1，P2にそれぞれ記憶されているエントリの一例を示した図である。パソコンP1にはフォルダAが記憶されており、その中にファイルa1，a2，a3，a4が格納されている。パソコンP2にはフォルダBが記憶されており、その中にファイルa5，b1，b2，b3，b4が格納されている。

【0133】

図１３は、図１２に示すファイルのうち，a1，a2，a3，a4，a5，b1，b2，b4がエントリ指定手段により指定され，更に，a1，a2，a3，a4，a5を一つのグループとしてエントリαとし，b1，b2，b4を一つのグループとしてエントリβとしている．エントリb3は指定されなかったエントリである．エントリαとエントリβからなるグループを新たなグループとして，エントリγとしてグループ定義手段により定義した例である．エントリに対する識別子を表す関数をideとし，ハッシュ関数に対する識別子を表す関数をidhとする．エントリa1，a2，a3，a4，a5に対して適用するハッシュ関数をhaとし，b1，b2，b4に対して適用するハッシュ関数をhb，エントリα，β，γに対するハッシュ関数をhとする．エントリ及びハッシュ関数に対する識別子は，エントリに対するハッシュ値と連接可能なデジタルデータとする．デジタルデータを‖を挟んで並べた表現は，それらのデジタルデータを連接したデジタルデータを表すものとする．

この例に対して，要約ファイル表現の例を示す．この例では，エントリ毎の要約ファイル表現として，該エントリのハッシュ値，該ハッシュ値を作成するのに用いたハッシュ値作成手段の識別子，及び該エントリを特定する識別子の連接により得られたデジタルデータを用いる．このとき，エントリa1，a2，a3，a4，a5に対する要約ファイル表現は，それぞれ，

ha(a1)‖ide(a1)‖idh(ha)，
ha(a2)‖ide(a2)‖idh(ha)，
ha(a3)‖ide(a3)‖idh(ha)，
ha(a4)‖ide(a4)‖idh(ha)，
ha(a5)‖ide(a5)‖idh(ha)

となる．また，エントリb1，b2，b4に対する要約ファイル表現は，それぞれ，

hb(b1)‖ide(b1)‖idh(hb)，
hb(b2)‖ide(b2)‖idh(hb)，
hb(b4)‖ide(b4)‖idh(hb)

となる．エントリαとエントリβに対する要約ファイル表現は，それぞれ，

h(ha(a1)‖ide(a1)‖idh(ha)‖ha(a2)‖ide(a2)‖idh(ha)‖
ha(a3)‖ide(a3)‖idh(ha)‖ha(a4)‖ide(a4)‖idh(ha)‖
ha(a5)‖ide(a5)‖idh(ha))‖
ide(α)‖idh(h)‖
ha(a1)‖ide(a1)‖idh(ha)‖
ha(a2)‖ide(a2)‖idh(ha)‖
ha(a3)‖ide(a3)‖idh(ha)‖
ha(a4)‖ide(a4)‖idh(ha)‖
ha(a5)‖ide(a5)‖idh(ha)

と

h(hb(b1)‖ide(b1)‖idh(hb)‖hb(b2)‖ide(b2)‖idh(hb)‖
hb(b4)‖ide(b4))‖
ide(β)‖idh(h)‖
hb(b1)‖ide(b1)‖idh(hb)‖

hb(b2) ‖ ide(b2) ‖ idh(hb) ‖
hb(b4) ‖ ide(b4) ‖ idh(hb)

になる．エントリγに対する要約ファイル表現は図１４に示すようになる。

【0134】

〔第４の実施形態〕

以下、本発明の第４の実施形態に係るバルク型時刻認証方法について説明する。尚、本実施形態は、上記第３の実施形態の変形例であるため、相違部分のみを説明する。

【0135】

本実施形態では、第３の実施形態で作成されたエントリγに対する要約ファイル表現に所定の附加データを加えてサマリファイル SF3 が生成される。

【0136】

サマリファイルSF3のハッシュ値を、SF3 をバイト列等のデジタルデータとして扱い、そのようなデジタル・データに対して適用するMD5 や SHA-1 等のハッシュ関数を適用した結果としてもよい。サマリファイル SF3 のハッシュ値をこのように決定した場合には、このサマリファイルのハッシュ値に対する時刻認証証明書 TST3 を用いて、時刻証明の対象となった１つのデジタル・データ、例えば a1 に対する時刻認証証明の検証をおこなうには、サマリファイル自体 SF3 を用いる必要がある。

【0137】

別の方法として、SF3に含まれるエントリγの要約ファイル表現から、 エントリγのハッシュ値を表す部分

h(h(ha(a1) ‖ ide(a1) ‖ idh(ha) ‖ ha(a2) ‖ ide(a2) ‖ idh(ha) ‖
ha(a3) ‖ ide(a3) ‖ idh(ha) ‖ ha(a4) ‖ ide(a4) ‖ idh(ha) ‖
ha(a5) ‖ ide(a5) ‖ idh(ha)) ‖
ide(α) ‖ idh(h) ‖
h(hb(b1) ‖ ide(b1) ‖ idh(hb) ‖ hb(b2) ‖ ide(b2) ‖ idh(hb) ‖
hb(b4) ‖ ide(b4)) ‖
ide(β) ‖ idh(h))

を抽出し、サマリファイルSF3のハッシュ値としてもよい。サマリファイル SF3 のハッシュ値をこのように決定した場合には、このサマリファイルのハッシュ値に対する時刻認証証明書 TST3 を用いて、時刻証明の対象となった１つのデジタル・データ、例えば a1 に対する時刻認証証明の検証は、a1の内容であるデジタル・データ、及びエントリ a1 に対する個別化サマリファイルを用いて以下のように実行することが出来る。ここで、エントリ a1 に対する個別化サマリファイルは、検証対象である a1 に係わるデータ

ha(a1) ‖ ide(a1) ‖ idh(ha)

と、検証対象である a1の兄弟の関係にあるエントリに係わるデータ

ha(ai) ‖ ide(ai) ‖ idh(ha)　(i = 2, ..., 5)

と、親に当たるエントリαの識別子ide(α)とそのハッシュ関数の識別子 idh(h)，親の兄弟に当たるエントリβにかかわるデータ

h(hb(b1) ‖ ide(b1) ‖ idh(hb) ‖ hb(b2) ‖ ide(b2) ‖ idh(hb) ‖
hb(b4) ‖ ide(b4)) ‖ ide(β) ‖ idh(h))、

及び親の親に当たるエントリγのし識別子 ide(γ) とそのハッシュ関数の識別子 idh(h) からなる。親の兄弟βの子供にあたるエントリb1，b2，b4 に係わるデータは含まれないため、上記のサマリファイル SF3 に比べてデータ量が小さくなる。

【0138】

一般に、時刻証明の対象となるファイルの総数がｎのときのサマリファイルのデータ量は n に比例するが、その中の一つのエントリに対する個別化サマリファイルは，それら複数のエントリのグルーピングを適切に行うことにより、 $\log_m(n)$ のオーダのデータ量とすることができる(ここで m はグルーピングにおける各グループの要素の数である)。従って、個別化サマリファイルを用いる検証法は、n が大きい場合に、サマリファイルを用いる検証法と比較して、検証に必要なデータの量を著しく小さくすることが出来ると

いう利点を持つ。

【0139】

a1 に対する個別化サマリファイルを用いて時刻認証証明の検証を行うため、まず、a1 の内容であるデジタル・データに所定のハッシュ関数 ha を適用して、ハッシュ値 ha(a1)を計算し、この値がa1 に対する個別化サマリファイル の a1 に係わるデータに含まれるハッシュ値と一致するか否かを検証する。

【0140】

上記の第１の検証が失敗したときは、a1 に対する時刻認証証明の検証は失敗する。

【0141】

上記の第1の検証が成功したとき、個別化サマリファイル利用型サマリファイル・ハッシュ値再計算手段により個別化サマリファイルからサマリファイル SF3 のハッシュ値を以下のように計算する。

【0142】

第1に、a1 に対する個別化サマリファイルからa1に係わるデータ

ha(a1) ‖ ide(a1) ‖ idh(ha)

を抽出する。

【0143】

第２に、前記当該のエントリa1に係わるデータ

ha(a1) ‖ ide(a1) ‖ idh(ha)

と、時刻証明の対象なるエントリの階層構造において当該のエントリ a1 と兄弟の関係にある1個以上のエントリ a2, a3, a4, a5 に係わるデータ

ha(a2) ‖ ide(a2) ‖ idh(ha),
ha(a3) ‖ ide(a3) ‖ idh(ha),
ha(a4) ‖ ide(a4) ‖ idh(ha),
ha(a5) ‖ ide(a5) ‖ idh(ha)

を連接し、該エントリa1 の親に当たるエントリ αに対するエントリ用ハッシュ関数 h を適用し、該エントリa1 の親に当たるエントリ αのハッシュ値

h(ha(a1) ‖ ide(a1) ‖ idh(ha) ‖
ha(a2) ‖ ide(a2) ‖ idh(ha) ‖
ha(a3) ‖ ide(a3) ‖ idh(ha) ‖
ha(a4) ‖ ide(a4) ‖ idh(ha) ‖
ha(a5) ‖ ide(a5) ‖ idh(ha))

を計算する。

【0144】

第３に、当該エントリa1 の親に当たるエントリ αの識別子 ide(α)、前記のαに対するハッシュ値、及びαに対するエントリ用ハッシュ関数 h の識別子 idh(h) を連接し、当該エントリa1 の親に当たるエントリ αに係わるデータ

h(ha(a1) ‖ ide(a1) ‖ idh(ha) ‖
ha(a2) ‖ ide(a2) ‖ idh(ha) ‖
ha(a3) ‖ ide(a3) ‖ idh(ha) ‖
ha(a4) ‖ ide(a4) ‖ idh(ha) ‖
ha(a5) ‖ ide(a5) ‖ idh(ha)) ‖
ide(α) ‖ idh(h)

を計算する。

【0145】

第４に、上記の当該エントリa1 の親に当たるエントリα係わるデータと、その兄弟にあたる βに係わるデータ

h(hb(b1) ‖ ide(b1) ‖ idh(hb) ‖ hb(b2) ‖ ide(b2) ‖ idh(hb) ‖
hb(b4) ‖ ide(b4)) ‖ ide(β) ‖ idh(h))

を連接して、その結果に最上位階層のエントリγに対するハッシュ関数 h を適用し、

サマリファイル SF3 のハッシュ値を計算する。

【0146】

上記は、階層構造が３段階の場合に、個別化サマリファイル利用型サマリファイル・ハッシュ値再計算手段がサマリファイルのハッシュを計算する手順を示しているが、階層構造が３段階より深くなる場合にも、ある時刻証明の対象となるエントリの個別化サマリファイルとして、以下の(1)及至(3) からなるデータを個別化サマリファイルとして用いることにより、個別化サマリファイル利用型サマリファイル・ハッシュ値再計算手段は当該のエントリを含むサマリファイルのハッシュ値を再計算することができる。

【0147】

(1) 当該エントリの内容であるデジタル・データ。

【0148】

(2) 当該エントリとその祖先に当たるエントリ（親、親の親、親の親の親等に当たるエントリ）に対する識別子とそのエントリに割り付けられたハッシュ関数の識別子。

【0149】

(3) 当該のエントリあるいはその祖先にあたるエントリの兄弟の関係にあるエントリに係わるデータ。

【0150】

〔その他〕

尚、上記実施形態１乃至４では、サマリファイル作成手段１１において、フォルダ管理手段１７で管理しておいた1個以上のデジタルデータｇの全てを使用して、サマリファイルＳＦを作成しているが、これに限らず、1個以上のデジタルデータｇの一部を使用してサマリファイルＳＦを作成してもよい。この場合、予め定められた指定方法により、サマリファイルＳＦに含めることを指定したエントリのみが時刻証明の対象となる。この指定方法としては、具体的なファイル名を指定する場合、ファイル名の拡張子を指定する場合、コンピュータとしての利用者端末１におけるファイルシステムが提供するエントリの属性（例えば、読み取り専用属性、隠しファイル属性、アーカイブ属性）の値により指定する場合がある。第３の実施形態は、時刻証明の対象となるエントリの指定する場合の一例である。

また、上記プログラム(p1),(p2),(p4),の各記録、インストール作業は、それぞれ利用者端末１，時刻認証装置２、検証装置４で読み取り可能な各プログラム(p1),(p2),(p4)が記録されているＣＤ－ＲＯＭ等の記録媒体を利用することによって行うことも可能である。

【図面の簡単な説明】

【0151】

【図1】本発明の第１の実施形態に係るバルク型時刻認証方法における登場主体の関係、通信インフラストラクチャ(Infrastructure)、及び各装置の構成を示した図。

【図2】サマリファイルＳＦの一部を表現した図。

【図3】サマリファイルＳＦの一部を表現した図。

【図4】サマリファイルＳＦの一部を表現した図。

【図5】サマリファイルＳＦの一部を表現した図。

【図6】サマリファイルＳＦ１全体を表現した図。

【図7】図６に示すサマリファイルＳＦ１に係るエントリの木構造を示した図。

【図8】図６に示すサマリファイルＳＦ１の作成行程を示したフロー図。

【図9】第１の実施形態におけるデータ通信処理を示したシーケンス図。

【図10】第２の実施形態におけるサマリファイルＳＦ２全体を表現した図。

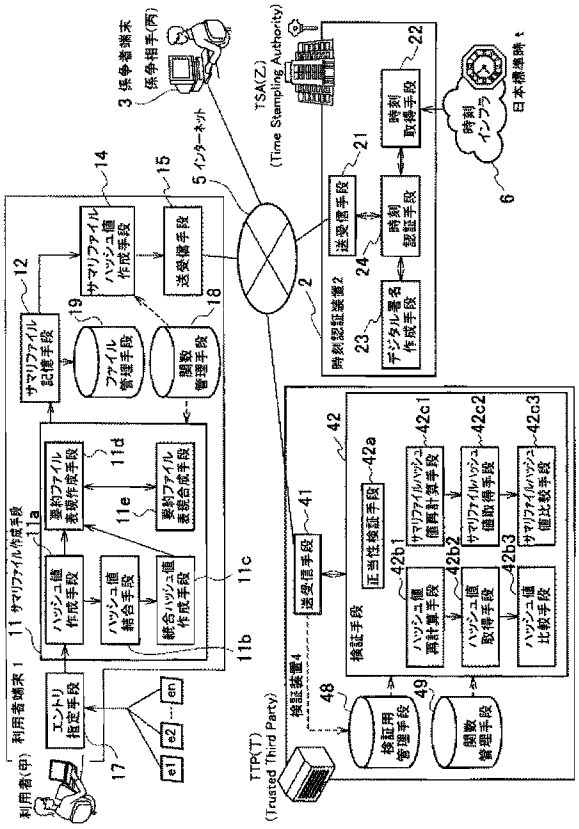
【図11】図１０に示すサマリファイルＳＦ２に係るエントリの木構造を示した図。

【図12】第３の実施形態を示し、ＬＡＮ(Local Area Network)やインターネット等の通信ネットワークで接続された２つのパソコン（パーソナル・コンピュータ）P1，P2と、これらパソコンP1，P2にそれぞれ記憶されているエントリの一例を示した図である。

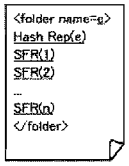
【図13】図１２に示すエントリに基づいて、新たに作成されたサマリファイルの木構造を示した図。

【図14】エントリァに対する要約ファイル表現を示した図。
【図15】従来の時刻認証方法における登場主体の関係、及び、通信インフラストラクチャを示した図。
【図16】従来のバルク型時刻認証方法における登場主体の関係、通信インフラストラクチャ、及び各装置の構成を示した図。
【図17】従来のバルク型時刻認証方法におけるデータ通信処理を示したシーケンス図。
【符号の説明】
【0152】
１　利用者端末
２　時刻認証装置
３　係争者端末
４　検証装置
５　インターネット
６　時刻インフラ
１１　サマリファイル作成手段
１１ａ　ハッシュ値作成手段
１１ｂ　ハッシュ値結合手段
１１ｃ　統合ハッシュ値作成手段
１１ｄ　要約ファイル表現作成手段
１１ｅ　要約ファイル表現合成手段
１２　サマリファイル記憶手段
１４　サマリファイルハッシュ値作成手段
１５　送受信手段
１７　フォルダ管理手段
１８　関数管理手段
１９　ファイル管理手段
２１　送受信手段
２２　時刻取得手段
２３　デジタル署名作成手段
２４　時刻認証手段
４１　送受信手段
４２　検証手段
４２ａ　正当性検証手段
４２ｂ１　ハッシュ値再計算手段
４２ｂ２　ハッシュ値取得手段
４２ｂ３　ハッシュ値比較手段
４２ｃ１　サマリファイルハッシュ値再計算手段
４２ｃ２　サマリファイルハッシュ値取得手段
４２ｃ３　サマリファイルハッシュ値比較手段
４８　検証用管理手段
４９　関数管理手段
h1　ファイル用ハッシュ関数（エントリ用ハッシュ関数の一例）
h2　統合用ハッシュ関数（エントリ用ハッシュ関数の一例）
h3　サマリファイル用ハッシュ関数

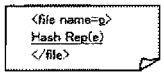
【図1】



【図2】

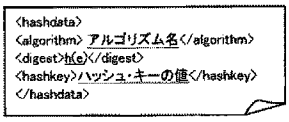


【図3】

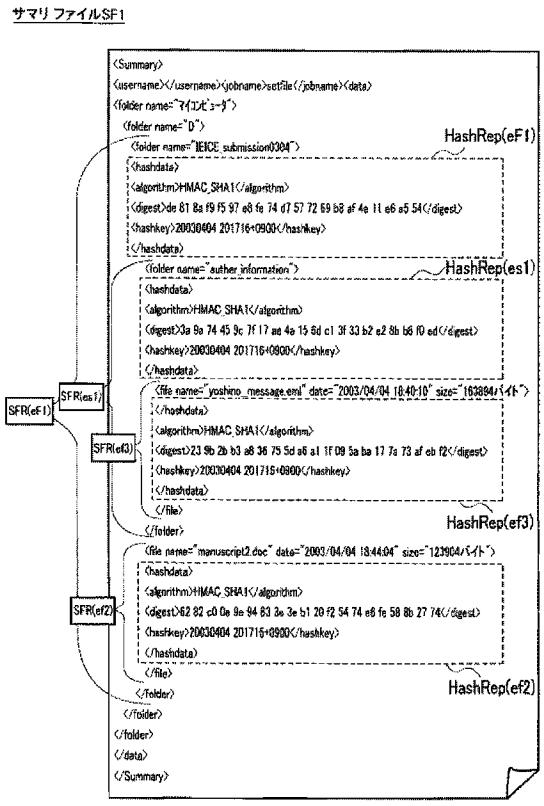


【図4】

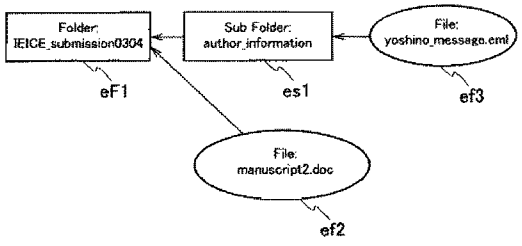
【図5】



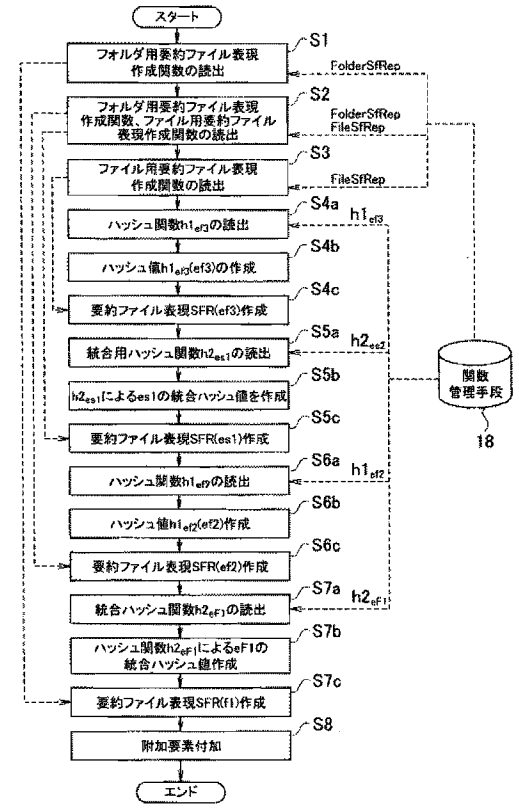
【図6】



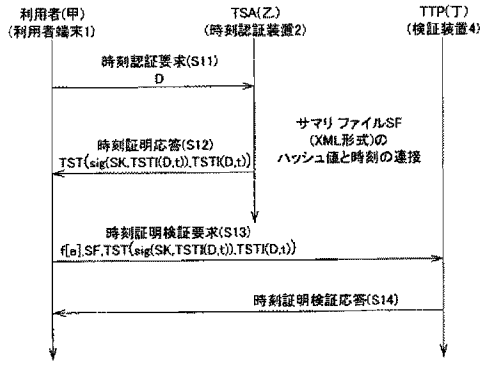
【図7】



【図8】

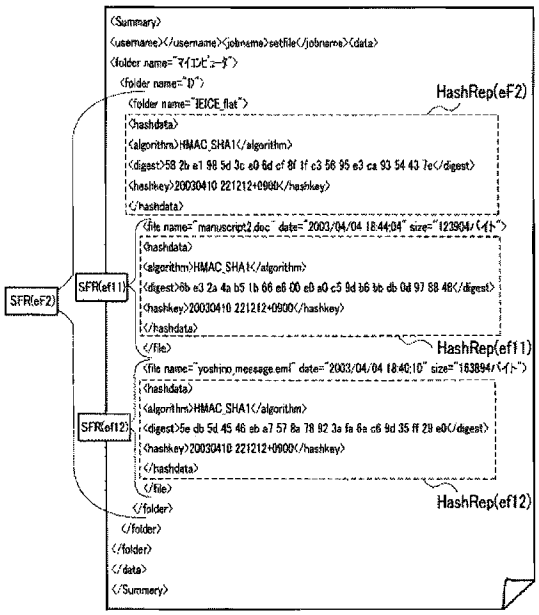


【図9】

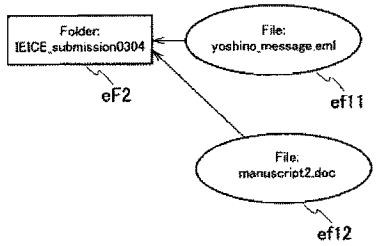


【図10】

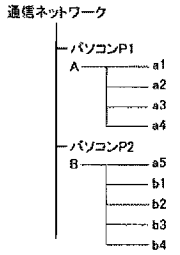
サマリ ファイルSF2(フラットな場合)



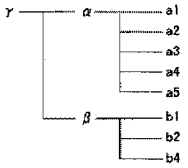
【図11】



【図12】



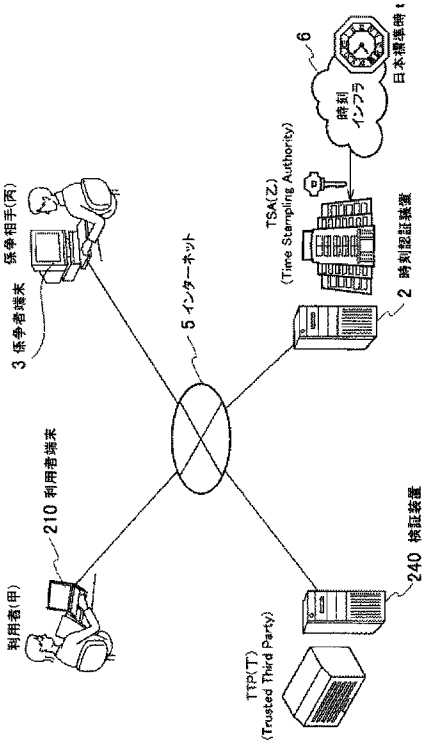
【図13】



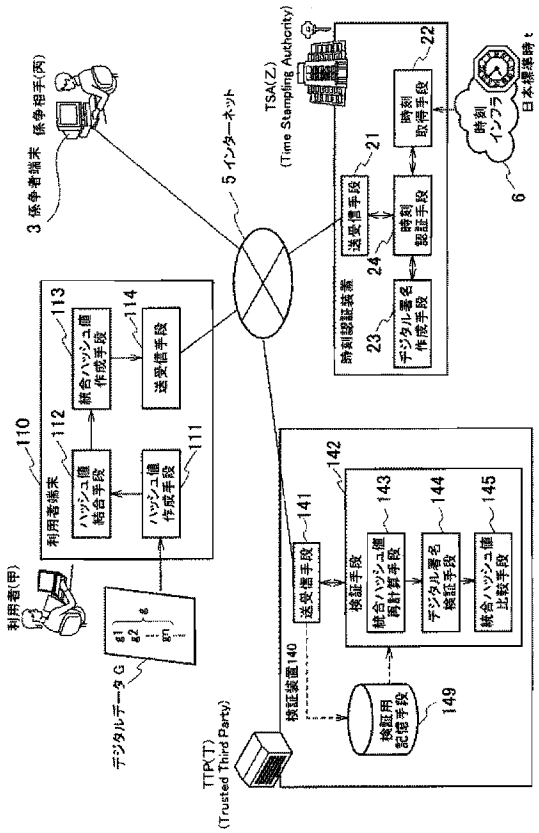
【図14】

```
h(h(ha(a1) ‖ ide(a1) ‖ idh(ha) ‖ ha(a2) ‖ ide(a2) ‖ idh(ha) ‖
    ha(a3) ‖ ide(a3) ‖ idh(ha) ‖ ha(a4) ‖ ide(a4) ‖ idh(ha) ‖
    ha(a5) ‖ ide(a5) ‖ idh(ha) ) ‖
  ide(α) ‖ idh(h) ‖
  h(hb(b1) ‖ ide(b1) ‖ idh(hb) ‖ hb(b2) ‖ ide(b2) ‖ idh(hb) ‖
    hb(b4) ‖ ide(b4) ) ‖
  ide(β) ‖ idh(h) ) ‖
ide(γ) ‖ idh(h) ‖
h(ha(a1) ‖ ide(a1) ‖ idh(ha) ‖ ha(a2) ‖ ide(a2) ‖ idh(ha) ‖
  ha(a3) ‖ ide(a3) ‖ idh(ha) ‖ ha(a4) ‖ ide(a4) ‖ idh(ha) ‖
  ha(a5) ‖ ide(a5) ‖ idh(ha) ) ‖
ide(α) ‖ idh(h) ‖
ha(a1) ‖ ide(a1) ‖ idh(ha) ‖
ha(a2) ‖ ide(a2) ‖ idh(ha) ‖
ha(a3) ‖ ide(a3) ‖ idh(ha) ‖
ha(a4) ‖ ide(a4) ‖ idh(ha) ‖
ha(a5) ‖ ide(a5) ‖ idh(ha) ‖
h(hb(b1) ‖ ide(b1) ‖ idh(hb) ‖ hb(b2) ‖ ide(b2) ‖ idh(hb) ‖
  hb(b4) ‖ ide(b4) ) ‖
ide(β) ‖ idh(h) ‖
hb(b1) ‖ ide(b1) ‖ idh(hb) ‖
hb(b2) ‖ ide(b2) ‖ idh(hb) ‖
hb(b4) ‖ ide(b4) ‖ idh(hb)
```

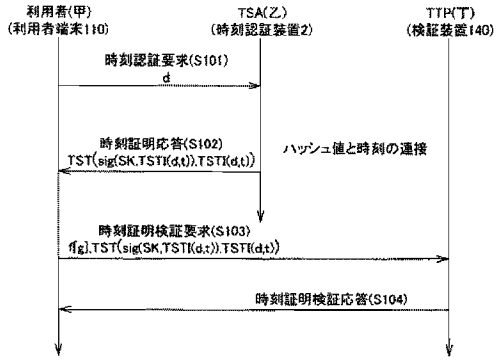
【図15】



【図16】



【図17】


利用者(甲)
(利用者端末110)
TSA(乙)
(時刻認証装置2)
TTP(丁)
(検証装置140)
時刻認証要求(S101)
d
時刻証明応答(S102)
TST(sig(SK,TSTI(d,t)),TSTI(d,t))
ハッシュ値と時刻の連接
時刻証明検証要求(S103)
f(g),TST(sig(SK,TSTI(d,t)),TSTI(d,t))
時刻証明検証応答(S104)